Orange
Innovation

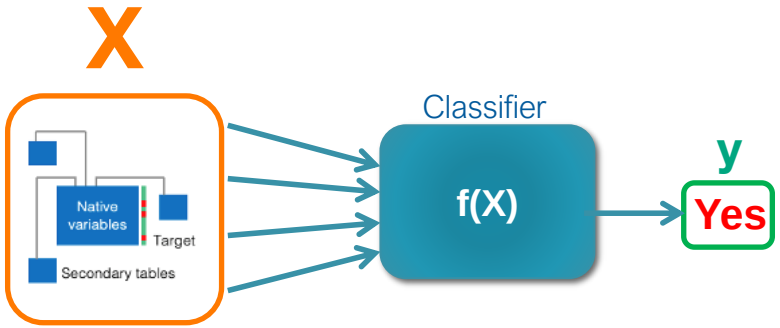
Khiops

an end-to-end solution for Automated Machine Learning

Nicolas Voisine
Marc Boullé

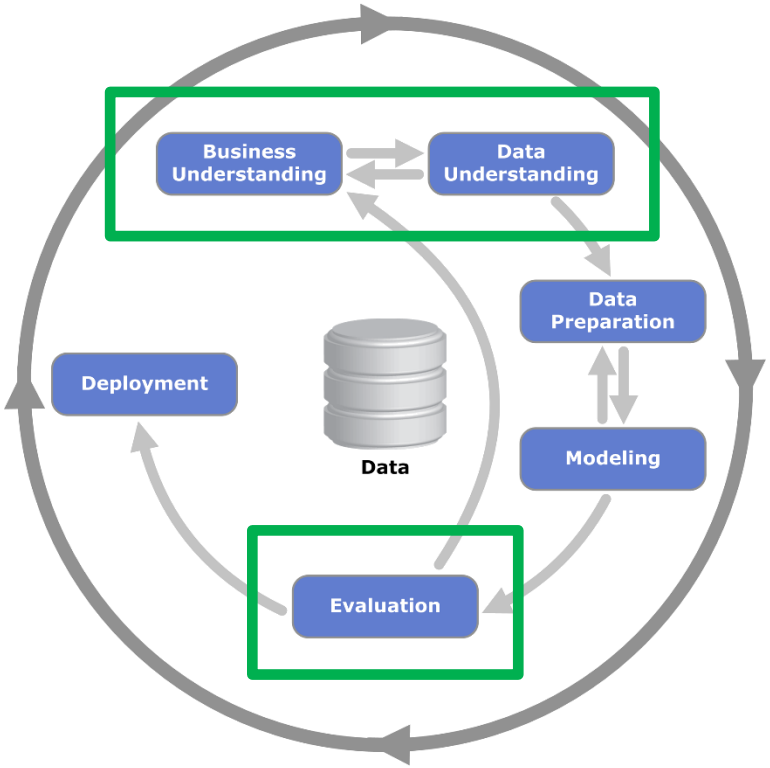
NormaSTIC 26 juin 2025

Orange needs to empower data experts (20 years ago)



1. **Business-Driven and Impact-Oriented:** Agnostic to use cases (e.g., detection of fraud, churn or anomalies), robust to poor quality data and focus on business understanding
2. **Production-Ready and Cost-Effective Scalability:** Built to process complex multi-table data structures efficiently and ensure reliable predictions in production environments
3. **Interpretability and robustness**: Ensured that models are interpretable and robust, enabling business experts to extract meaningful insights and undersand the story behind the data

Focus on data and business understanding, the rest is automated



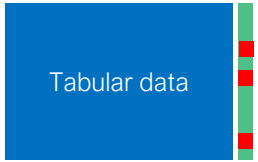
Most use cases don't justify allocating 30 guys for marginal improvements:

- Small improvements rarely influence real-world decisions and lever actions
- Over-processing just adds waste (muda) without return on investment

The green steps are for you; the rest is for the robots!

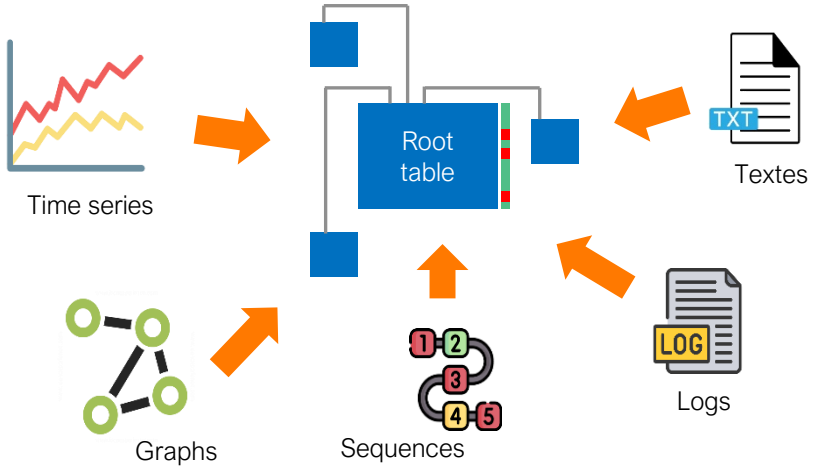
Tabular data in real-world

At school, we learn Data science with toy datasets



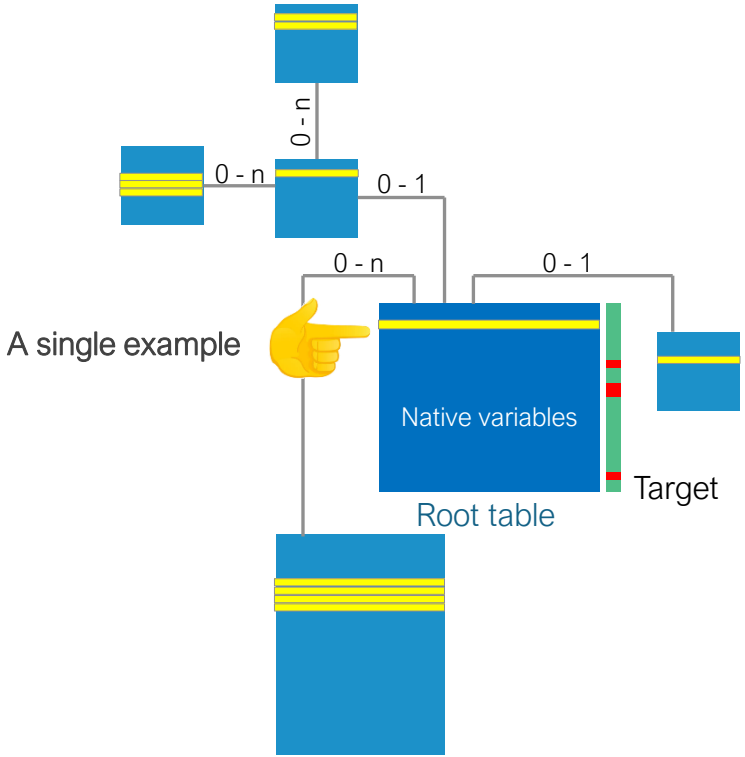
A statistical individual = A single row
Very easy !!

In reality, data is much more complex



Multi-table schema can encode diverse type of data

Khiops champion to handle complex data



Input | Output

Khiops **automates** the construction of aggregates (flattening), which **saves us time** and avoids **human bias** and **overfitting**

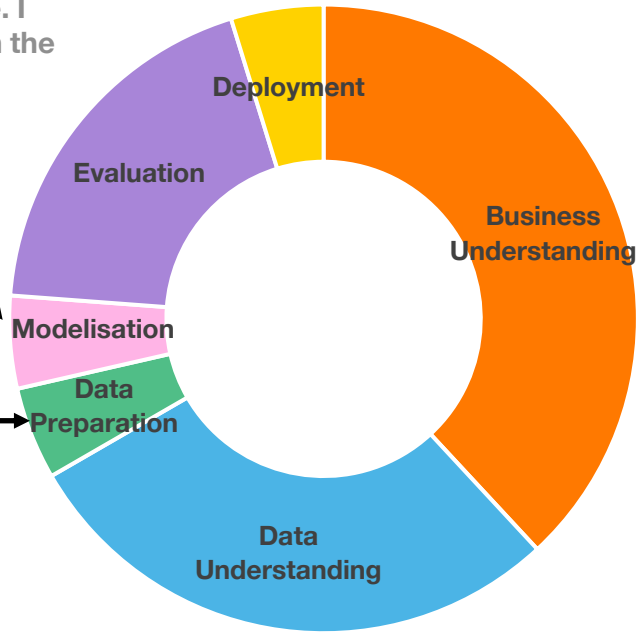
Exemple : $Count(Selection(\text{Usages}, YearDay(\text{useDate}) \text{ in } [1;90] \text{ and } \text{Product}=\text{"VOD"})$

Months of savings with Khiops

Standard CRISP distribution on multi-table data

Khiops delivers results in which I always have total confidence. I save time because I question the data rather the model.

Robustness, interpretability and efficiency facilitate inference



Distribution with Khiops

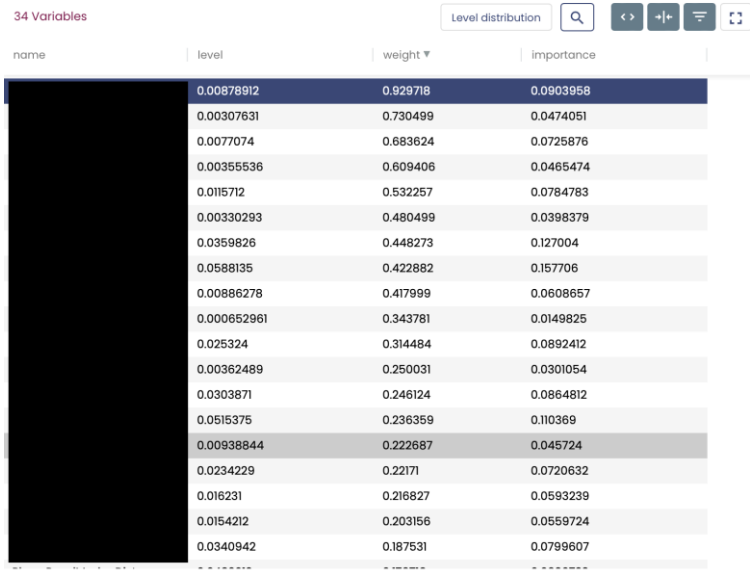
Khiops natively digest CDRs (relational data) and manages encoding and feature engineering

Model Interpretability for informed decision-making

Khiops results can be easily analyzed using its **built-in visualization** tool

The way each variable is linked to the target **is clear**

The final model selects only **few independant variables**



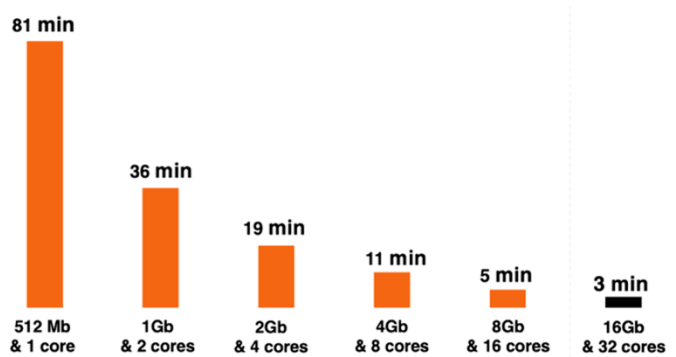
A master of frugality on big data

A single end-to-end formalism, no hyperparameters and low-level implementation for outstanding performance

- On a use case for used battery detection:
  Khiops was **50x faster** with **200x fewer resources** than Google AutoML
- On a benchmark based on 8 internal datasets, Khiops uses **50x fewer resources** than **Prevision.io** and **4x fewer** than **H20** for comparable prediction performance

It's **green**, it's **cheaper**, and it adapts easily to available resources: **from mono-thread** with little memory (out-of-core) to **multi-machines** (**Kubernetes** native)

Illustration:
Khiops run on our LiveBox (very limited memory)



Calculation time for a 9Gb dataset
(500000 instances, 2000 variables, 2 classes)

Khiops in 2025

https://khiops.org/

More than 130 scientific publications in national and internationals journals and conferences

State of the art software engineering:

- **400 000** lines of code C++ for the core part
- **10 000** lines Java code for the Desktop GUI
- **15 000** lines of code for khiops-python
- **800** sets of non-regression tests

- Evaluated with volumetric test up to:
  - Hundreds of millions of instances
  - Hundreds of thousands of variables
  - Hundreds of GB for data files

Short demo

We use the « as a service » version of Khiops that would be move to open source shortly

DEMO

NumAcc	An	Jour	...	Mortel
201800000001	18	1	...	False
201800000002	18	2	...	False
201800000003	18	3	...	False
...	...	...	...	

1:1

NumAcc	Cat Route	...	Ecole
201800000001	3	...	False
201800000002	4	...	False
201800000003	5	...	False
...	...	...	

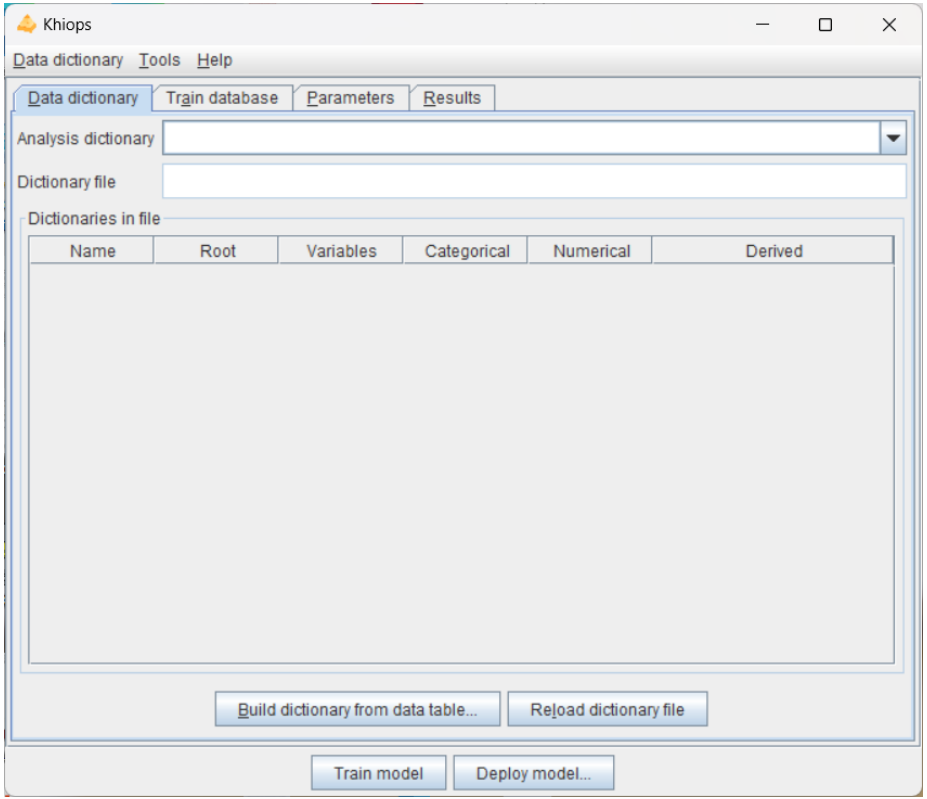
1:n

NumAcc	Num Veh	Sens circ	...	Manoeuvre
201800000001	B01	1	...	Sans chang.
201800000001	A01	1	...	Tourn. Droite
201800000002	A01	1	...	Sans chang.
...	...	...	...	

0:n

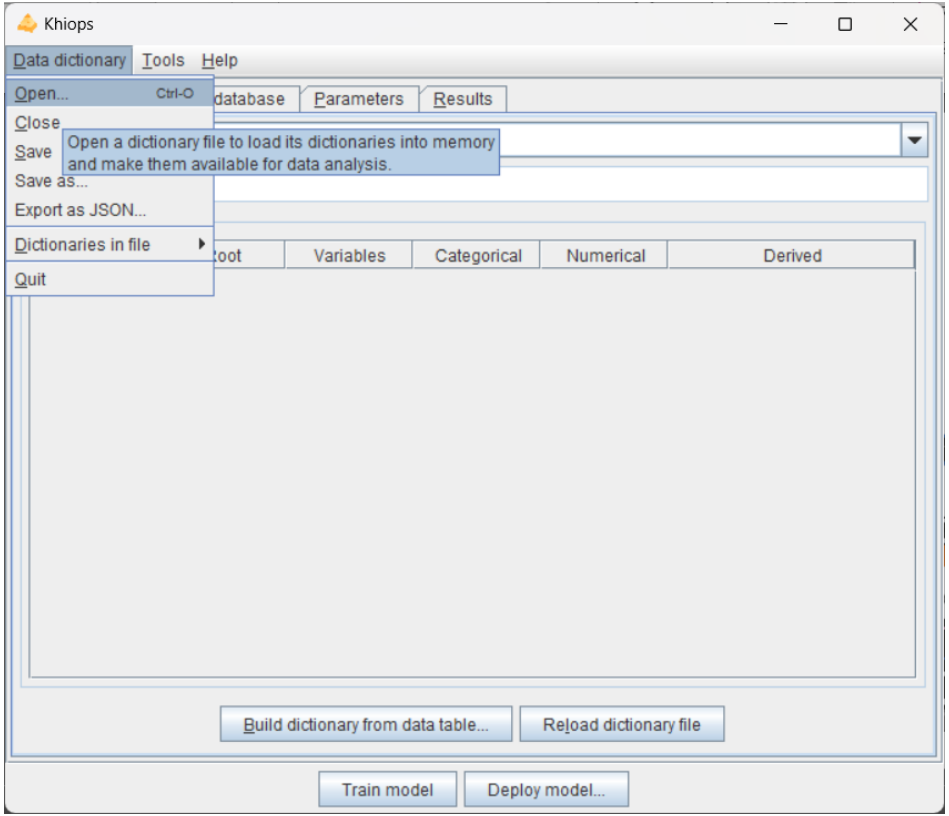
NumAcc	Num Veh	Place	...	Annee naissance
201800000001	B01	1	...	1928
201800000001	A01	1	...	1960
201800000002	A01	1	...	1947
...	...	...	...	

Short demo



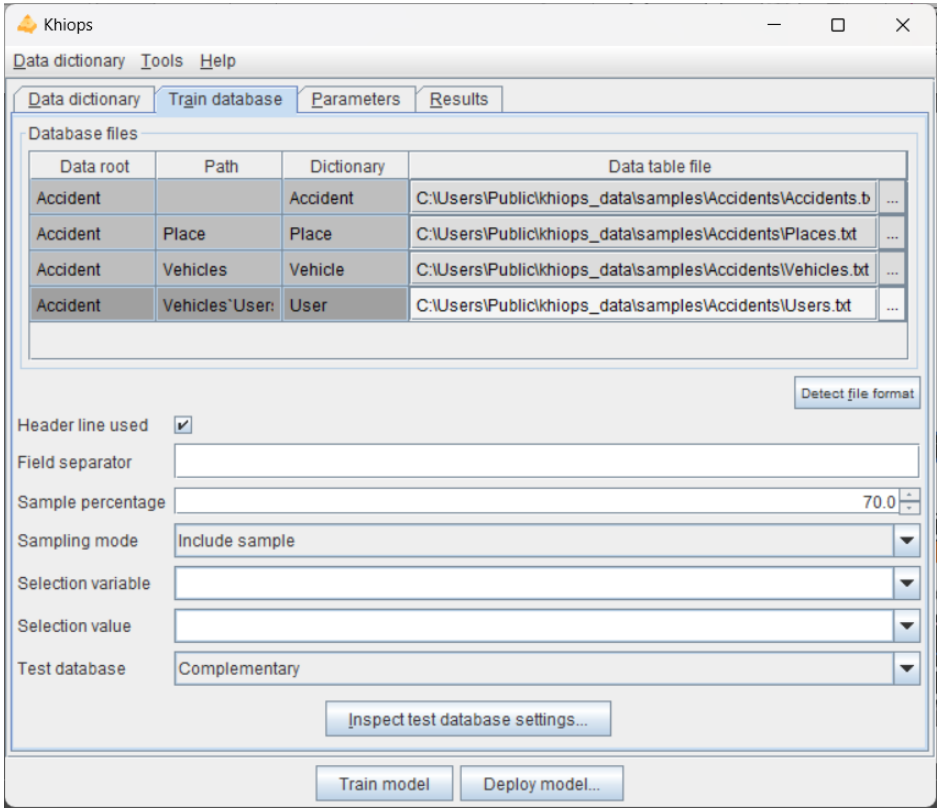
1 Open dictionary

Short demo



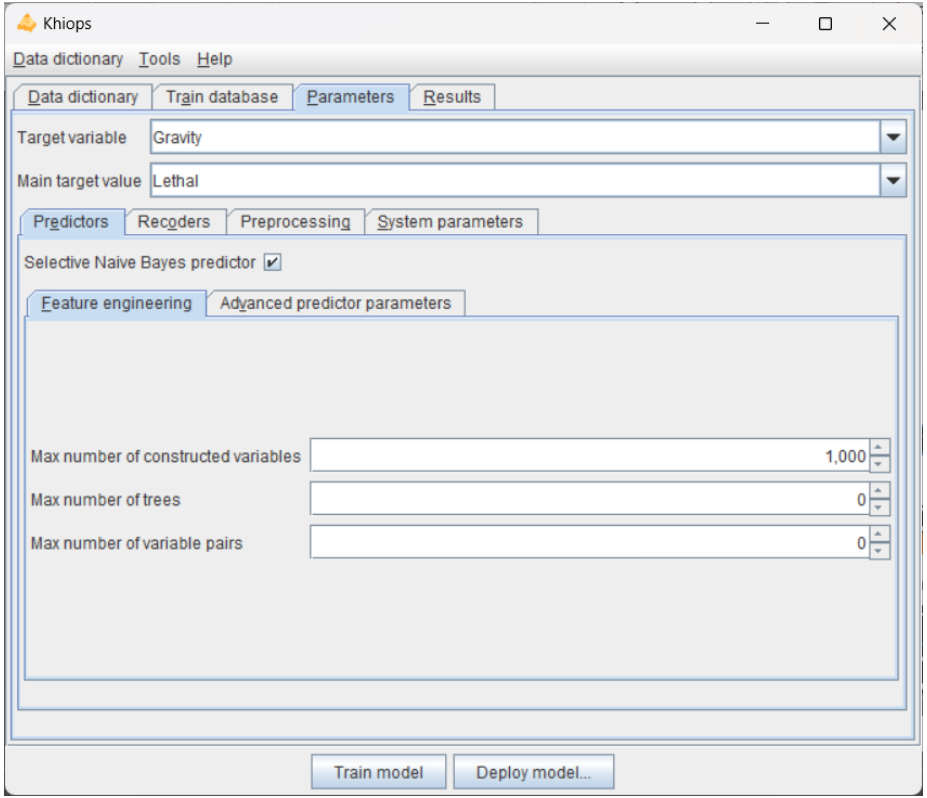
1 Open dictionary

Short demo



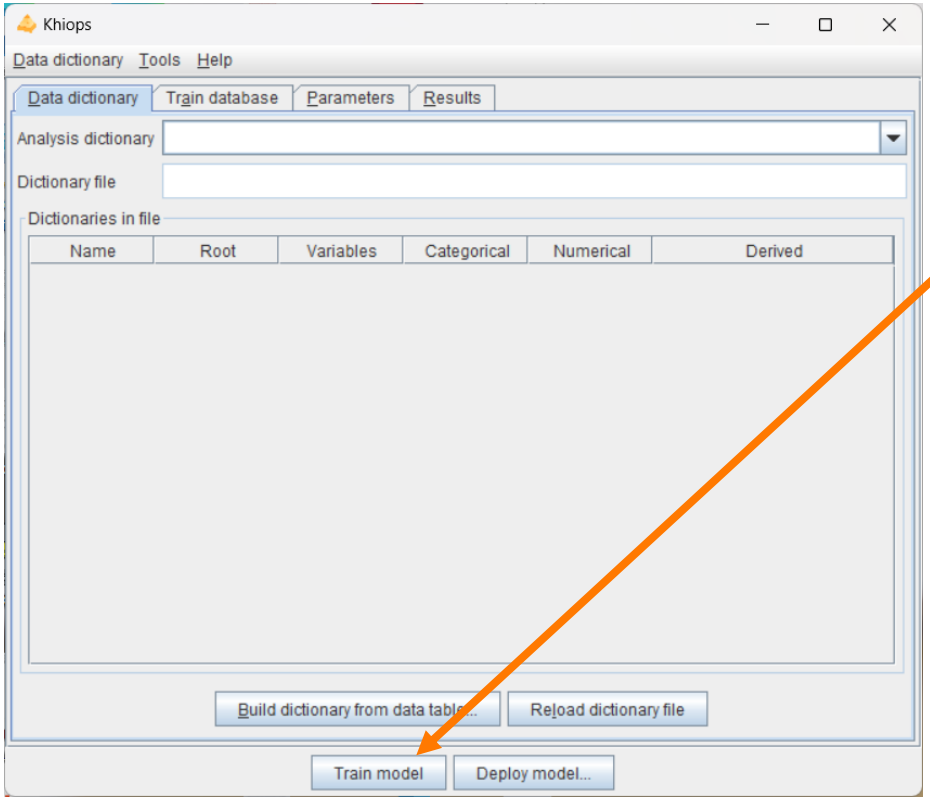
2 set data set files

Short demo



3 analysis configuration

Short demo



4 Run

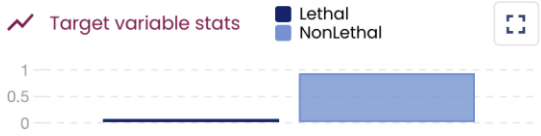
Summary

Dictionary: Accidents

Database: s3://pr-diod-ocp-fr01-khiops-as-a-service-prod-s3bucketclaim01/project/edc48638-babe-40e8-9f75-b61486799858/Accidents.txt_sorted

Target

Target variable stats



Trained predictors

Selective Naive Bayes | 54 Variables

Select trained predictor: Selective Naive Bayes

54 Variables

Level distribution

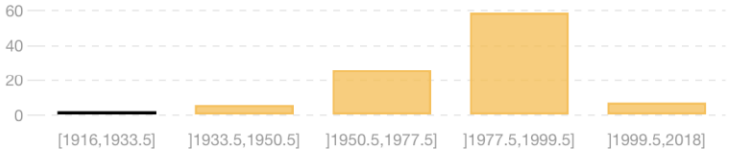
Name	Level	Weight	Importance
Light	0.023844	0.162598	0.0622655
Hour.DecimalTime	0.00597559	0.640625	0.0618717
Mode(Vehicless.Maneuver) where Categor	0.0259667	0.128906	0.0578556
Mean(Vehicless.Mean(Userss.BirthYear))	0.0164264	0.19043	0.0559292
Mode(Vehicless.Category) where Maneuve	0.0166102	0.146484	0.0493268
Mode(Vehicless.Category) where Direction	0.0109246	0.204956	0.0473188
Max(Vehicless.Min(Userss.BirthYear))	0.0119991	0.181641	0.0466855
Mode(Vehicless.Maneuver) where Maneuve	0.0307446	0.0644531	0.044515
Mode(Vehicless.ImpactPoint) where Categ	0.013172	0.138184	0.0426633
CountDistinct(Vehicless.Direction)	0.00363414	0.498047	0.0425437
Mode(Vehicless.Maneuver) where ImpactP	0.0108922	0.158203	0.0415112

Max(Vehicless.Min(Userss.BirthYear))

Scale chart

Distribution

Coverage

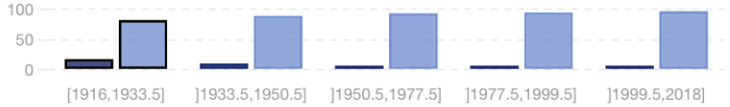


Target distribution

Lethal | NonLethal

Values

Probabilities



Current interval

Orange
Innovation

Khiops.org

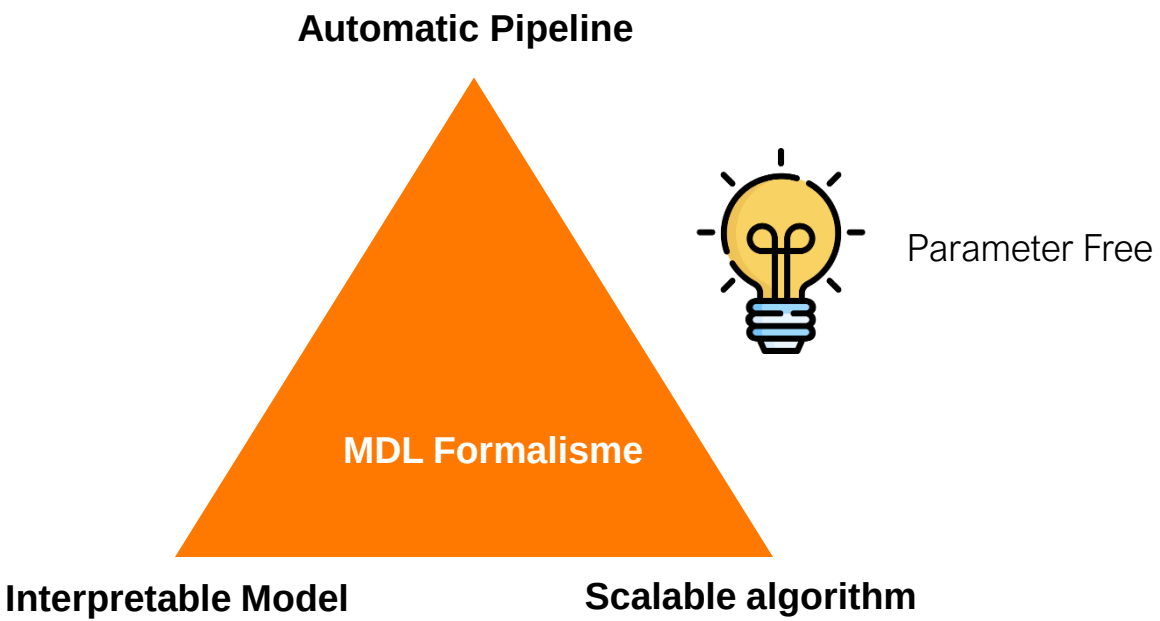
Conclusion

« Khiops est un outil productif,
qui s'adapte à beaucoup de cas d'usage,
et fait gagner énormément de temps dans les projets de data-science.»

Nous pouvons intervenir chez vous pour des ateliers d'accompagnement.
Contactez-nous à **khiops.team@orange.com**

orange™

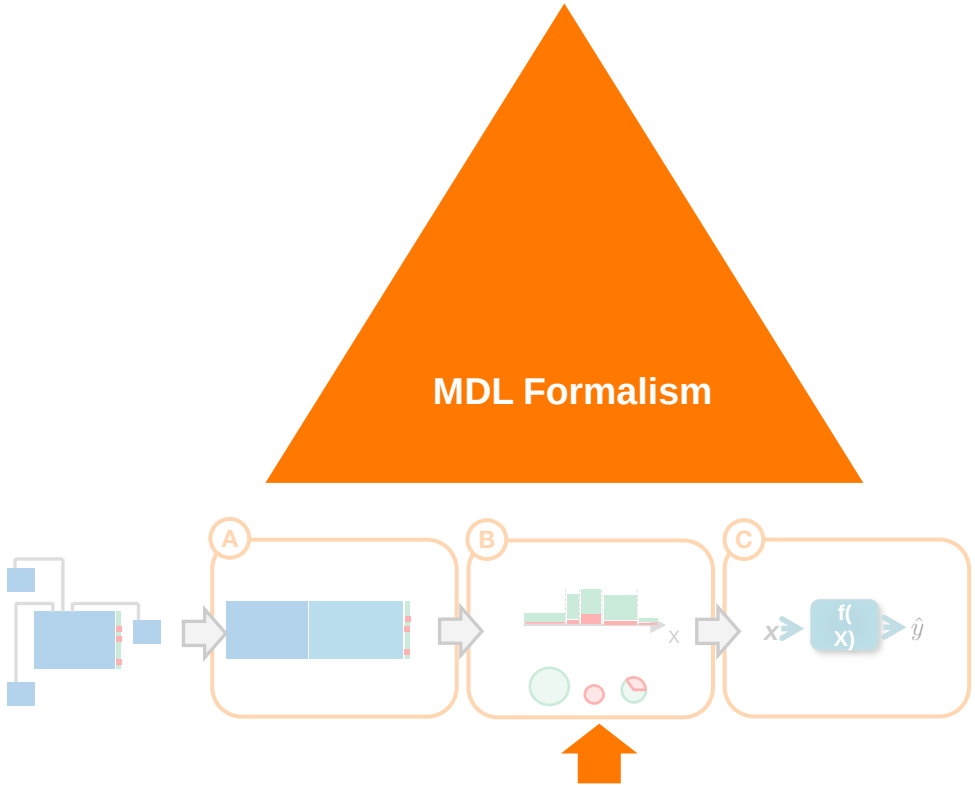
Khiops take advantage of unique formalism base on MDL*



* Minimum Description Length

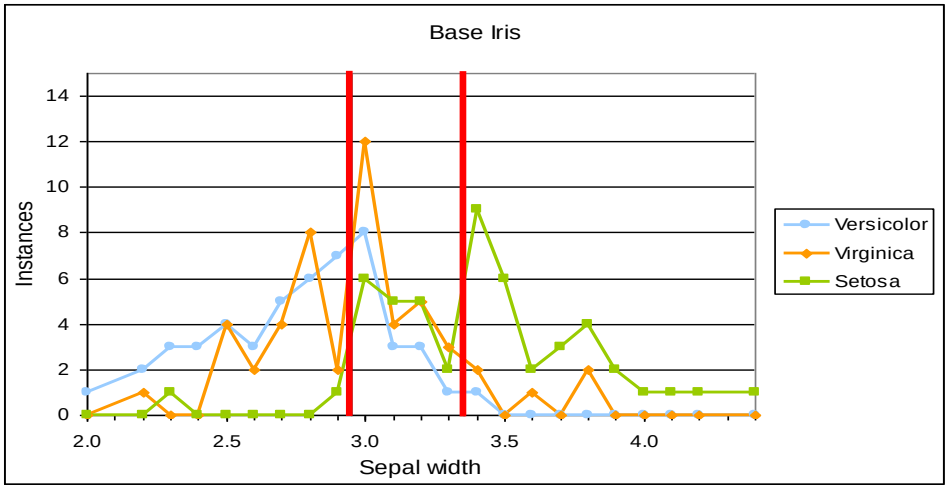
MDL is used end to end !

Optimal Encoding

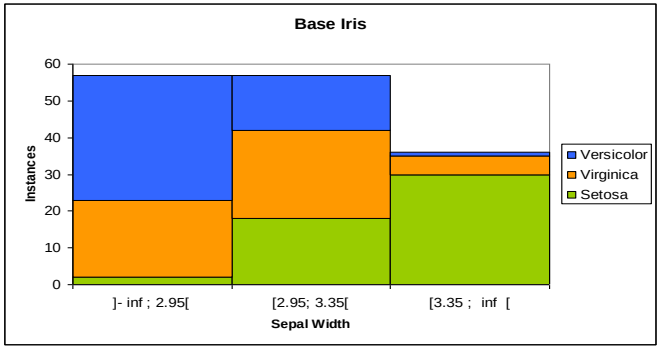


Formalisme MDL appliqué à la discrétisation

Étape B du pipeline : prétraitement univarié



Versicolor	34	15	1
Virginica	21	24	5
Setosa	2	18	30
Sepal width	]- inf ; 2.95[	[2.95; 3.35[	[3.35 ; inf [



Quel est le meilleur modèle ?

Formalisme MDL appliqué à la discrétisation

Une approche bayésienne de sélection de modèle

Sélection du modèle le plus probable connaissant les données:

$$\textit{ArgMax}\ P(M|D) = \frac{\overbrace{P(M)}^{\text{Prior}}\ \overbrace{P(D|M)}^{\text{Likelihood}}}{P(D)}$$

Formalisme MDL appliqué à la discrétisation

Un critère d'optimisation régularisé

$$\mathcal{C}(M) = \log N + \log \binom{N+I-1}{I-1} + \sum_{i=1} \log \binom{N_i+J-1}{J-1} + \sum_{i=1} \log \frac{N_i!}{N_{i1}!N_{i2}!\ldots N_{iJ}!}$$

Vraisemblance des données connaissant le modèle

Description de la distribution des classes

Choix de la position des bornes

Choix du nombre d'intervalles

Régularisation intrinsèque

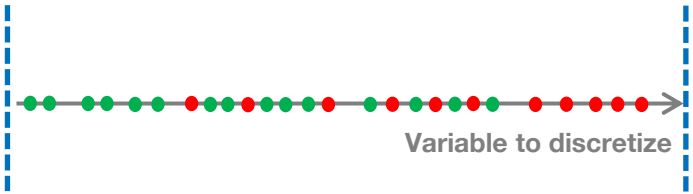
Formalisme MDL appliqué à la discrétisation

Équilibre entre *prior* et *vraisemblance* pour lutter contre le sur-apprentissage

$$\log N + \log \binom{N+I-1}{I-1} + \sum_{i=1} \log \binom{N_i+J-1}{J-1}$$

Optimisation du « Prior » seul

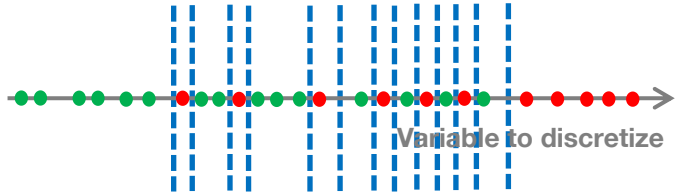
A single interval



$$\sum_{i=1} \log \frac{N_i!}{N_{i1}!N_{i2}!\dots N_{iJ}!}$$

Optimisation de la « vraisemblance » seule

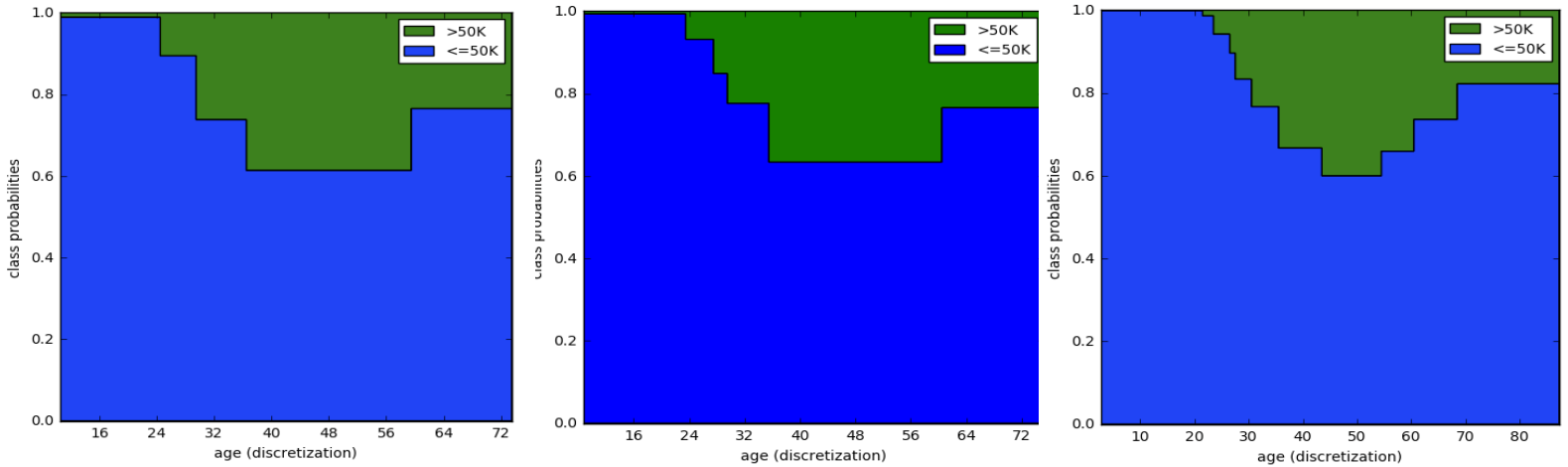
On interval for each « pure » zone



Formalisme MDL appliqué à la discrétisation

Le point d'équilibre dépend de la taille du jeu de données

L'exemple de la variable « Age » du dataset « Adult »

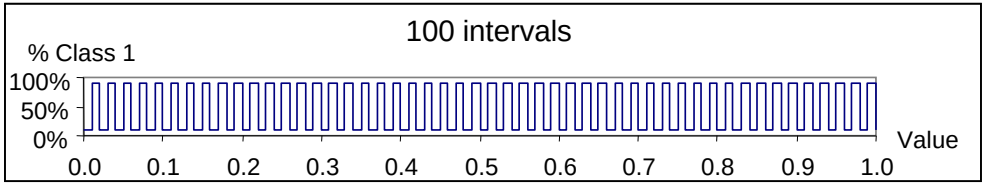


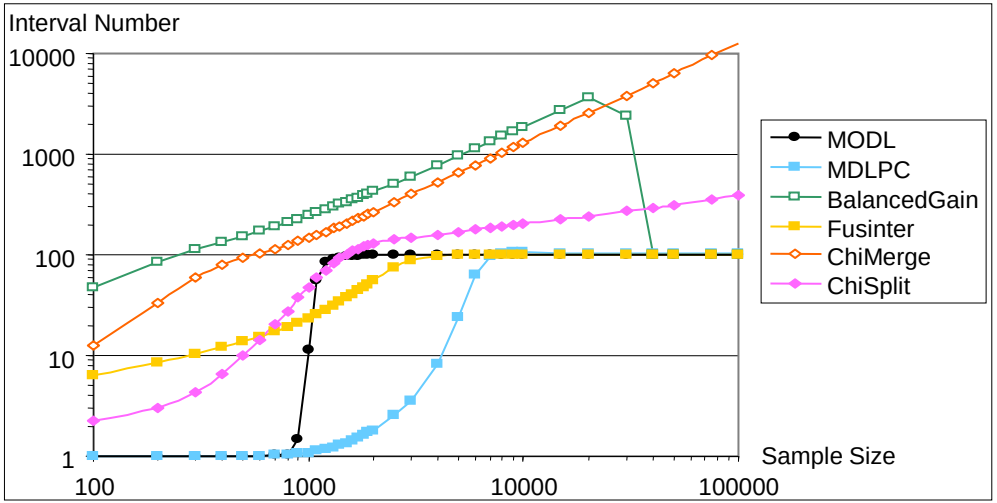
N / 2 N N x 2

Source : Adult Dataset, https://archive.ics.uci.edu/dataset/2/adult Modèles de discretization appris par Khiops – 2023

Univariate preprocessing : optimization criterion for the discretization

The example of the crenel pattern



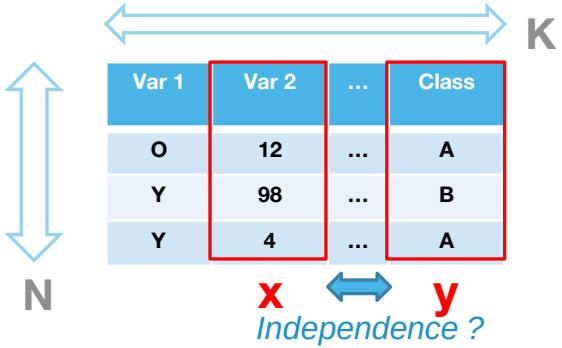


MODL correctly identifies the relevant information with a minimal number of instances

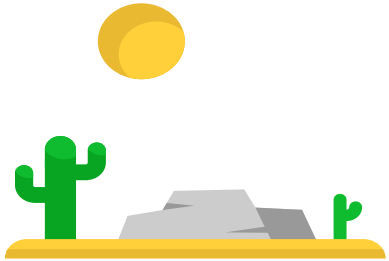
Variable Selection : compression gain based filtering

Why we want to select variables ?

1 – Scaling of the learning algorithms



2 – Accuracy of the learned models



Curse of dimensionality

In the literature, there are two families of variable selection approaches :

- **envelope** : The learned classifier is used to select the useful variable (ex: threshold on features importance)
- **filter** : Variables are selected before learning the classifier

Variable Selection : compression gain based filtering

What is compression gain ? (or **level** in the Khiops interface)

$$CG = 1 - \frac{-\log(P(M \mid D))}{-\log(P(M_0 \mid D))}$$

Model to be evaluated

Null model that includes a single interval / group

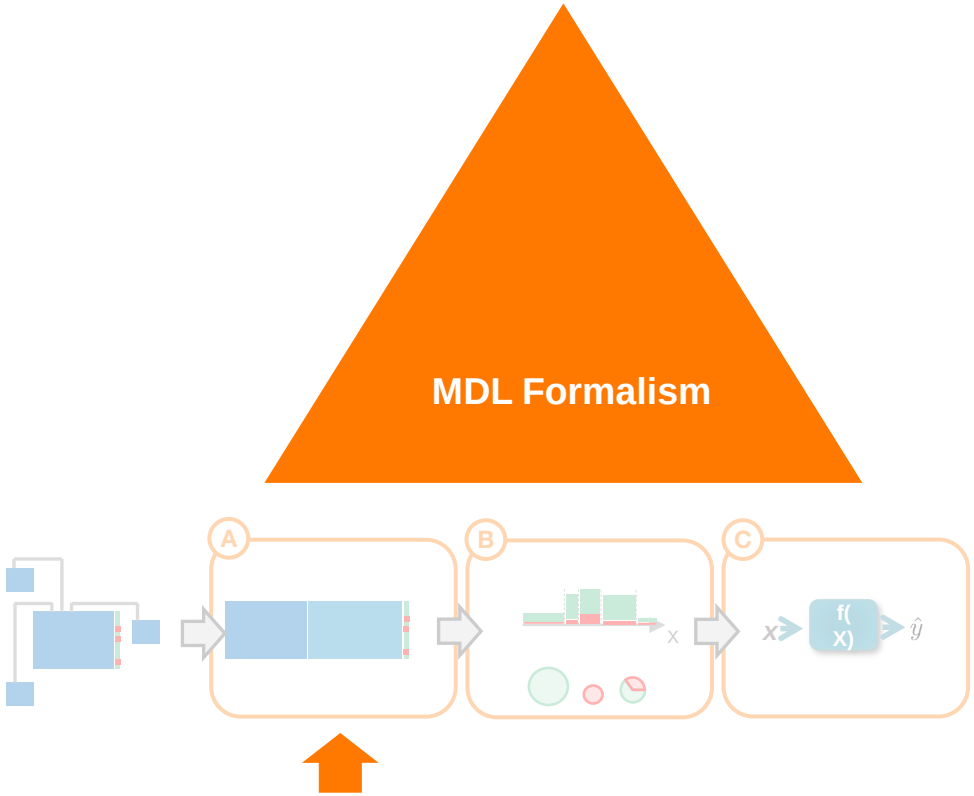
Ratio of coding length

- **Filtering method :** We keep only the variables with a positive compression gain

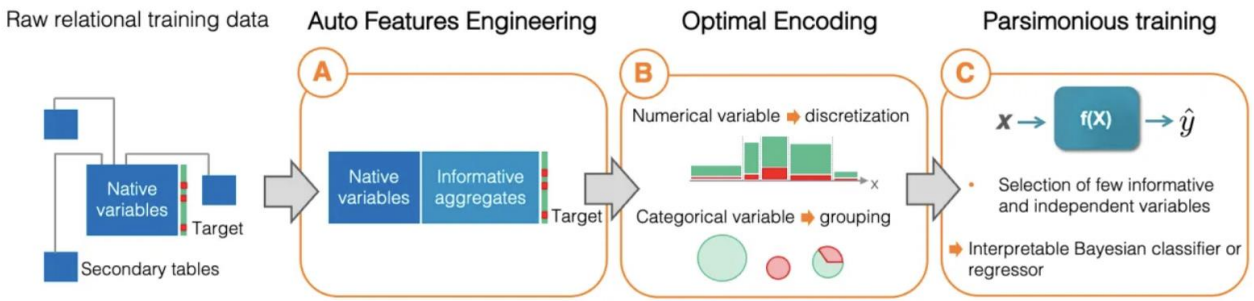
- **Intuition** : We remove variables for which the optimal model is less probable than the null model

MDL is used end to end !

Auto Features Engineering



MODL and Auto Features Engineering

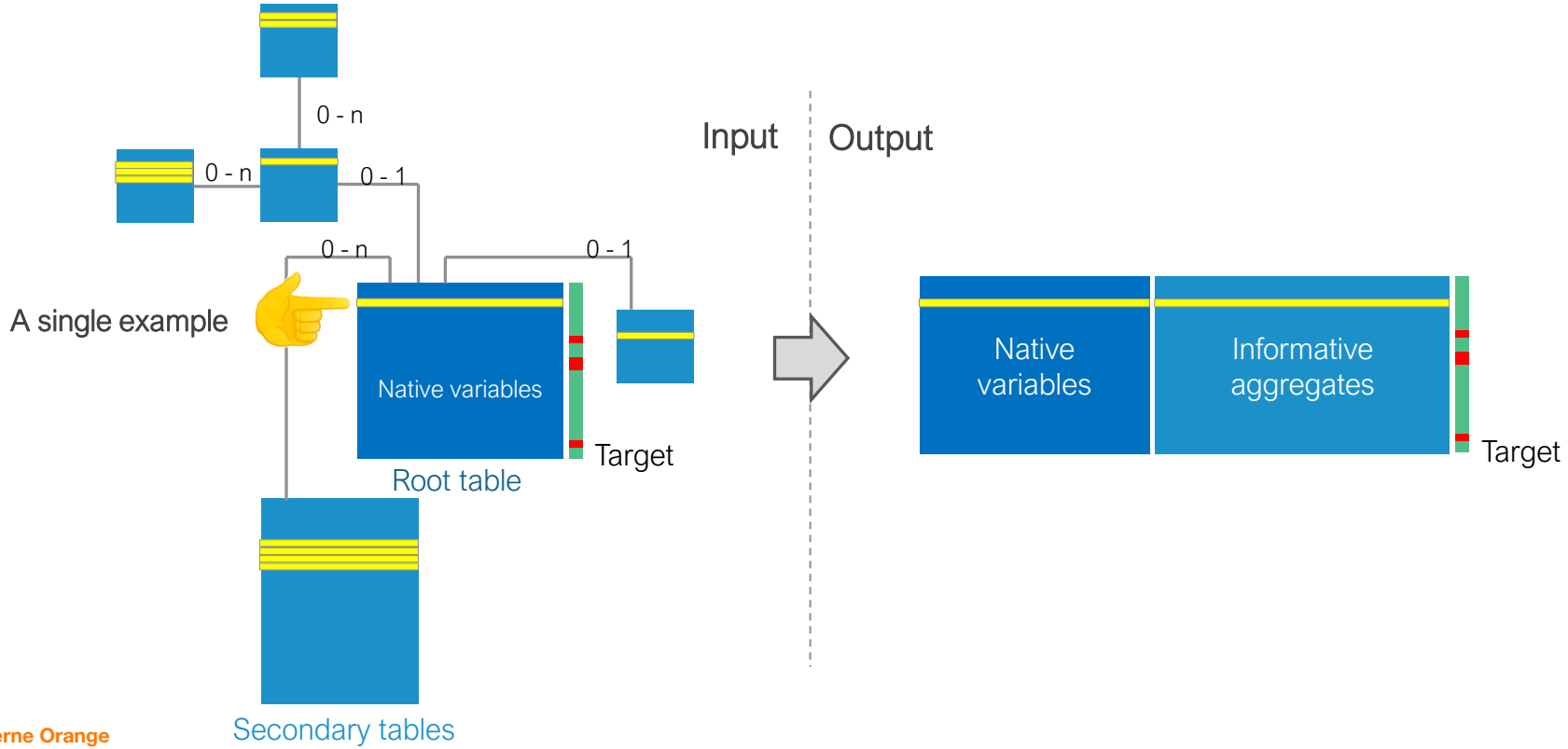


1 **A new prior with building cost**

2 **An efficient sampling algorithm for features generation**

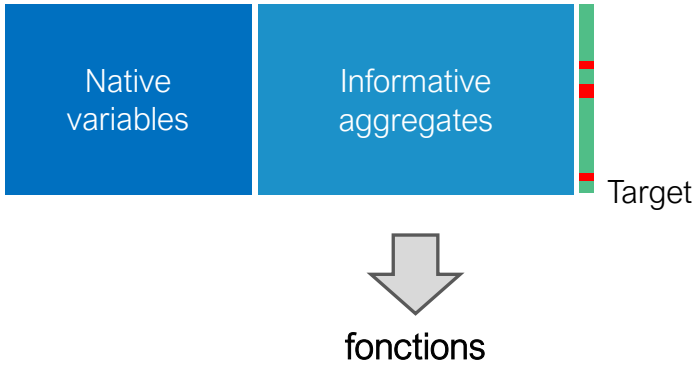
Auto Feature Engineering save time

Avoids time consuming of data-scientists



Auto Feature Engineering genrate important productivity gain

Aggregates generation



Regularization:
Aggregates simpler are more probable

Aggregates are interpretable, here some examples:

1. $Mode$(Usages, Product)
2. Max(Usages, $YearDay$(useDate))
3. $Count(Selection$(Usages, $YearDay$(useDate) in [1;90] and Product="VOD"))

1 - A new prior

- Generic expression for discretization and grouping criteria:

$$c(X) = L(M_P(X)) + L(D_Y \mid M_P(X), D_X)$$

Coding length: *describe the model* *encode the target by using the model*

- The multi-table approach implies and additional term within the prior:

$$c(X) = L(M_C(X)) + L(M_P(X)) + L(D_Y \mid M_P(X), D_X)$$

construction cost
which penalizes complex aggregate features

1 - A new prior

- The new prior has a **recursive** form due to function composition *(f ∘ g)*

$$L(M_C(X)) = \log(K+1) + \log R + \sum_{op \in \mathcal{R}} L(M_C(X_{op}))$$

Choice of constructing a new feature

for each operand

choice of the construction rule

choice of constructing a new feature

- Case of original features

$$L(M_C(X)) = \log K$$

1 - A new prior

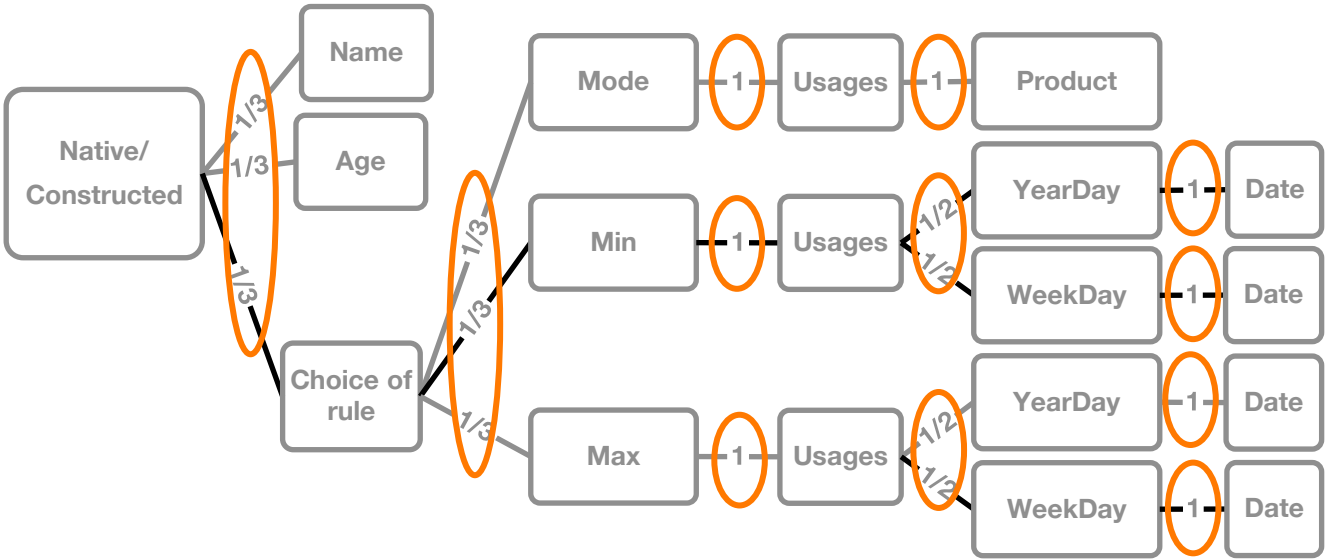
- **What's appends with variable filtering ?** 

  - Variables are filtered after the Auto Feature Engineering step *(univariate discretization and grouping)*
  - Compression gains take into account the new prior for constructed variables

The filtering step favors **simple and informative** constructed variables ☺

2 – An efficient sampling algorithm

- **Objective :** draw K constructed / native variables from this prior distribution
- This prior can be represented by a tree structure
- And it consists of a **H**ierarchy of **M**ultinomial **D**istributions with potentially **I**nfinite **D**epth *(HMDID)*



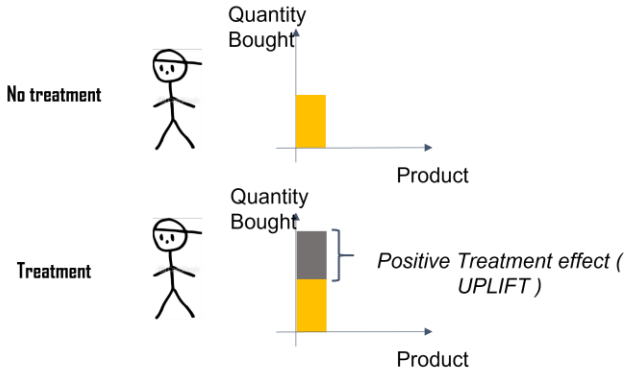
Khiops V11

- Major version
  - Text data
  - SNB classifier for sparse data
  - Shapley
  - Random forests for regression
  - Khiops interpretation
  - Histograms
  - Coclustering instances x variables
  - New visualization tools
  - Simplified ergonomy

UMODL an MODL approch for Uplift

Mina Rafla, Nicolas Voisine et Bruno Cremilleux

Uplift



- Uplift modeling[1] aims at estimating the change in an individual's behavior caused by a treatment.
- Y : output variable, T : treatment and X : features
  Real Uplift for each individual = $Y(T = 1) - Y(T = 0)$
- Both values cannot be known simultaneously.
- So, we consider two groups: treatment and control groups.

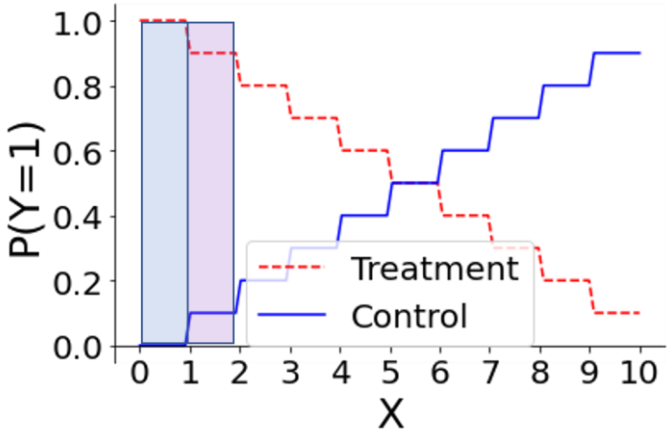
$$CATE : \hat{\tau}_n = \mathrm{E}[Y_n(T = 1)|X_n] - \mathrm{E}[Y_n(T = 0)|X_n] \quad (1)$$

Motivation

Motivation: detect areas with homogeneous treatment effect.

- To enable us to estimate uplift for each individual
- Know feature importance

Idea: creating a discretization method that separates areas with different outcome densities in treatment and control groups.



UMODL Discretization

Proposed method: **UMODL** a method based on the MODL approach[2] and designed for uplift.

M : an uplift discretization model. D : data. From a Bayes point of view:

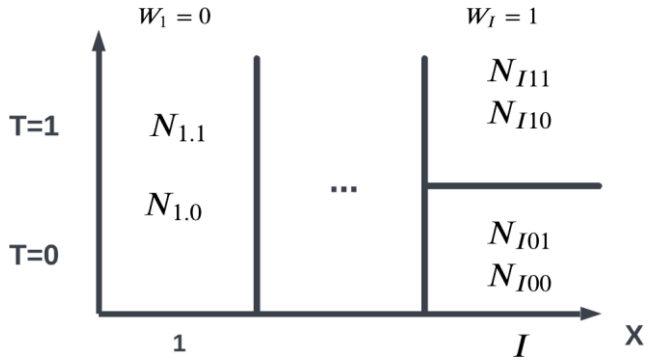
$$P(M \mid D) = \frac{P(M)P(D \mid M)}{P(D)}$$

- $P(M)$: prior
- $P(D \mid M)$: likelihood
- $P(D)$: constant

Objective: Maximize $P(M \mid D)$, hence maximize $P(M)P(D \mid M)$ → Tradeoff between prior and likelihood

UMODL Discretization

We define an uplift discretization model M by the hierarchy of parameters:



- I : number of intervals
- N_i : number of instances in the interval i
- W_i : boolean term indicating if the treatment has an effect in interval i (W_i=1) or not (W_i=0)
- $\{N_{i.j}\}_{W_i=0}$: number of instances in the interval i of class j
- $\{N_{itj}\}_{W_i=1}$: number of instances in the interval i of class j and the treatment t

UMODL Discretization

By exploiting the hierarchy of parameters and assuming the independence of the local distributions for each interval:

$$P(M)P(D|M) = P(I) \times P(\{N_i\}|I) \times \prod_i P(W_i|I)\,[(1-W_i) \times P(\{N_{i.j}\}|I,\{N_i\}) + W_i \times \prod_t P(\{N_{itj}\}|I,\{N_{it.}\})] \times \prod_i P(D_i|M)$$

A discretization model is Bayes optimal if its cost is minimal. Model's cost $C(M)$:

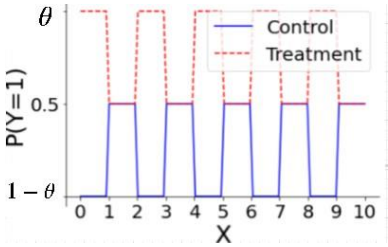
$$C(M) = -\log(P(M)P(D|M)) = \log N + \log\binom{N+I-1}{I-1} + I \times \log 2 + \sum_{i=1}^{I}(1-W_i)\log\binom{N_i+J-1}{J-1} + \underbrace{\sum_{i=1}^{I}(1-W_i)\log\frac{N_i!}{N_{i.1}!..N_{i.J}!}}_{Likelihood} + \sum_{i=1}^{I} W_i \sum_t \log\binom{N_{it.}+J-1}{J-1} + \underbrace{\sum_{i=1}^{I} W_i \sum_t \log\frac{N_{it.}!}{N_{it1}!..N_{itJ}!}}_{Likelihood}$$

A greedy search algorithm is implemented to find the parameters that minimize $C(M)$.

UMODL Evaluation

Goal: Evaluate whether UMODL is a good uplift estimator

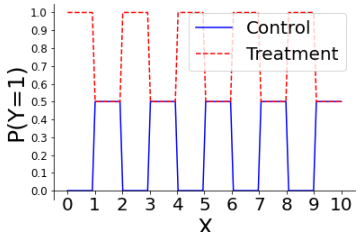
1. Define synthetic data patterns, for example:



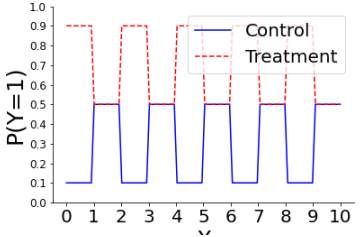
UMODL Evaluation

Goal: Evaluate whether UMODL is a good uplift estimator

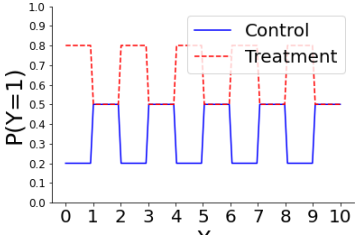
1. Define synthetic data patterns, for example:



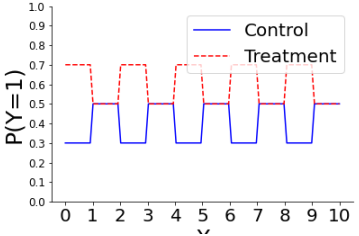
$\theta = 1$



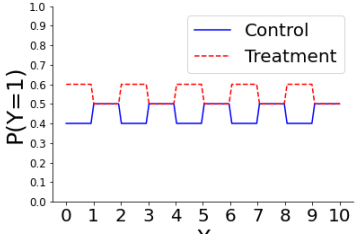
$\theta = 0.9$



$\theta = 0.8$



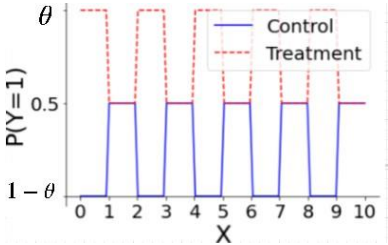
$\theta = 0.7$



$\theta = 0.6$

UMODL Evaluation : protocol

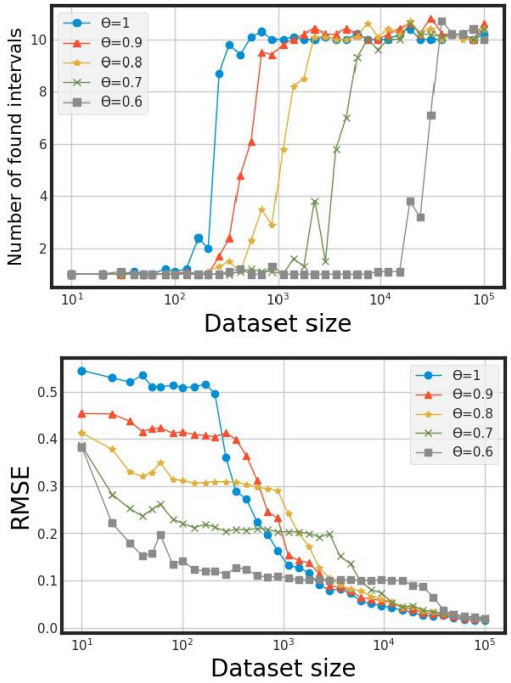
1. Define synthetic data patterns, for example:



2. Generate several datasets:
   - different sizes from 10 to 10,000 instances.
   - data is uniformly distributed on $[0, 10]$ for $T = 1$ and $T = 0$.
3. Generate a test set of 10K examples for the same uplift pattern.
4. Apply UMODL to find the Bayes optimal model.
5. Test the model by comparing the RMSE for each data point between:
   - the CATE estimation in the found interval and
   - the real CATE value.
6. Observe the number of found intervals and the RMSE, how do they behave with the data size.

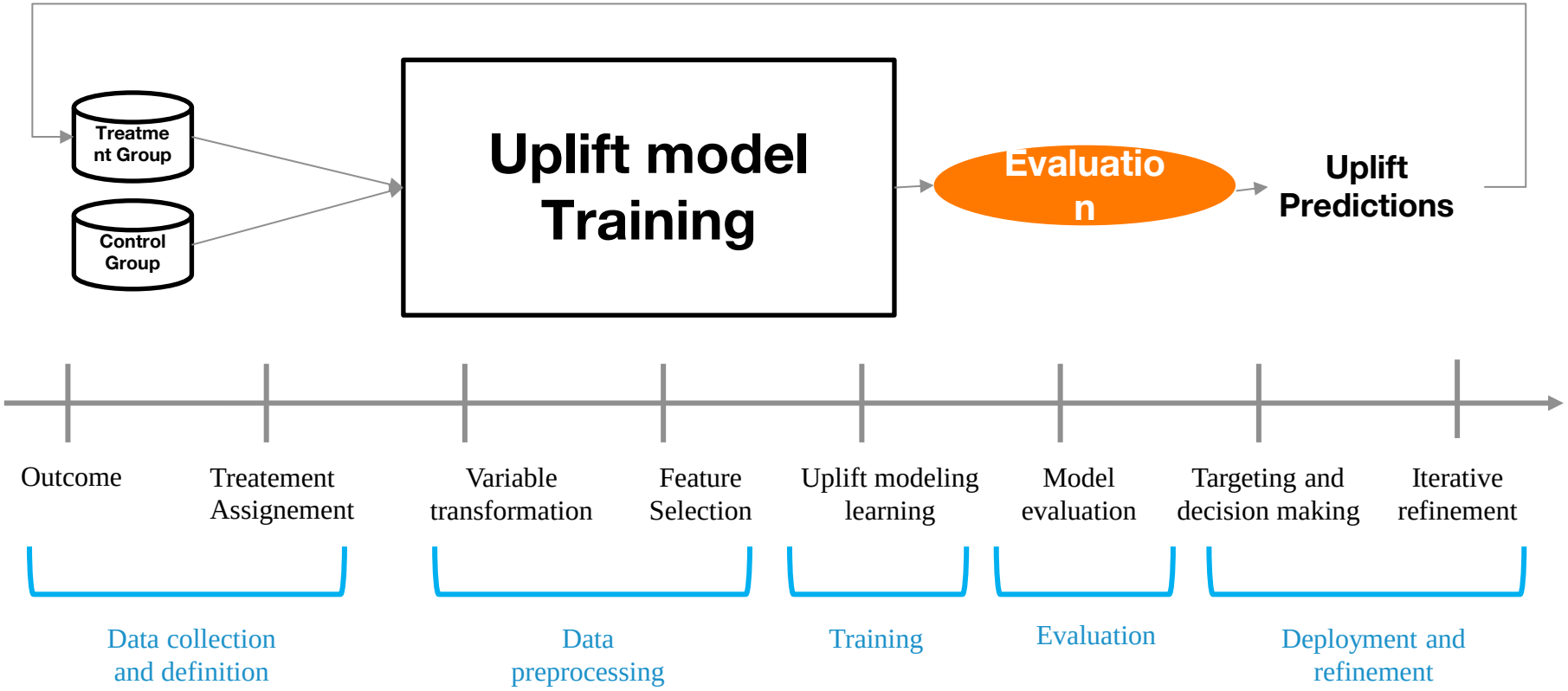
Other patterns were tested as well!

UMODL Evaluation : results



- **UMODL is a good estimator of uplift**: uplift data patterns are found (i.e correct number of intervals and a RMSE close to 0.)
- **Number of instances**: when the difference of densities between intervals gets smaller, UMODL needs more data to consider a separation to two intervals.
- **UMODL does not overfit**: when the data size increases significantly, UMODL does not consider extra intervals.

Uplift Modeling Life Cycle



KUPLIFT Package Python

https://github.com/UData-Orange/kuplift

1. **Feature selection using UMODL-FS:**

```
Import kuplift as kp
fs = kp.FeatureSelection()
important_vars = fs.filter(df[column_names], df["treatment"], df["outcome"])
```

2. **Variable transformation using UMODL:**

```
Import kuplift as kp
ue = kp.UnivariateEncoding()
encoded_data = ue.fit_transform(df[column_names], df["treatment"], df["outcome"])
```

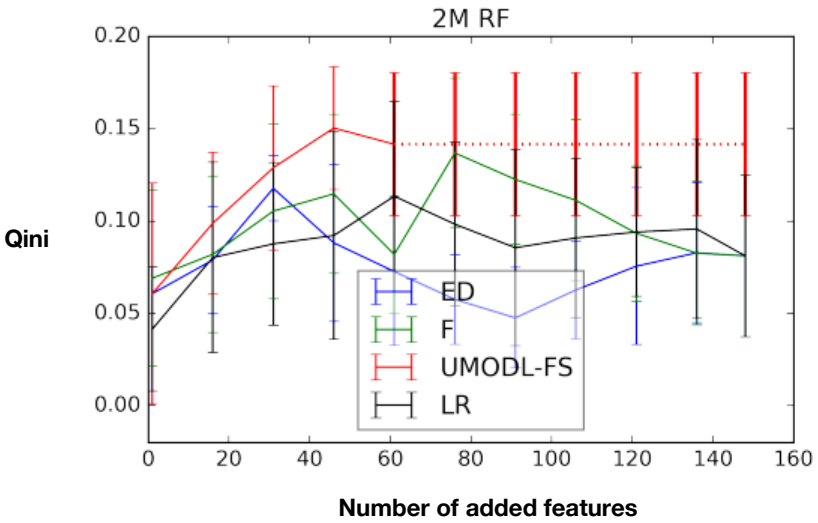
1. **Decision trees using UB-DT:**

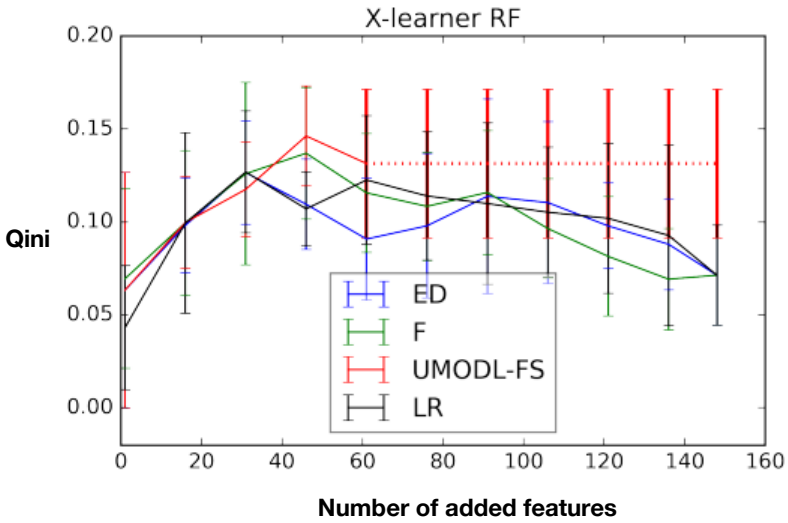
```
tree = kp.BayesianDecisionTree()
tree.fit(df[column_names], df["treatment"], df["outcome"])
preds = tree.predict(df[column_names])
```

2. **Random forests using UB-RF:**

```
forest = kp.BayesianRandomForest(n_trees=4)
forest.fit(df[column_names], df["treatment"], df["outcome"])
preds = forest.predict(df[column_names])
```

KUPLIFT Package Python





Merci