Explainable machine learning for survival analysis

Hubert Baniecki, Mateusz Krzyziński, Mikołaj Spytek,
Przemysław Biecek

MI^2.AI,
Warsaw University of Technology & University of Warsaw,
Warsaw, Poland

SLDS Survival Focus Group, LMU Munich
November 11th, 2022

About us

Shoutout to David Rügamer!
for connecting with us on ECML PKDD 2022

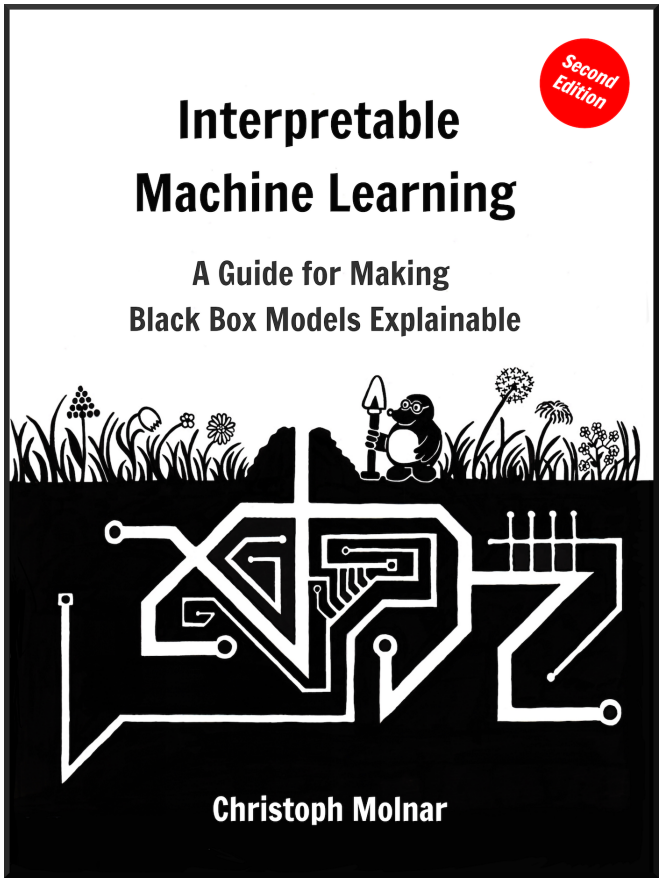


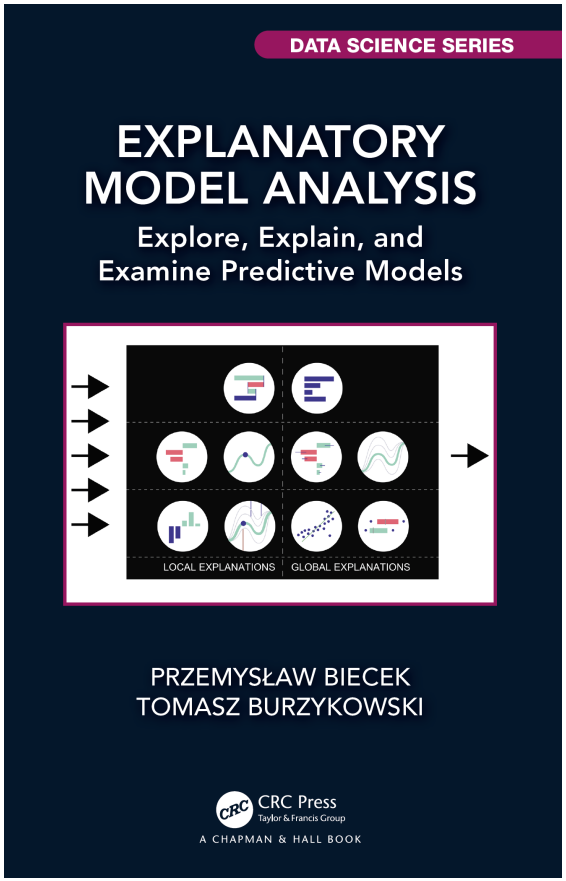

MI RESEARCH

MI^2.AI research lab, Warsaw, Poland
Topics: explainable machine learning & computational biomedicine

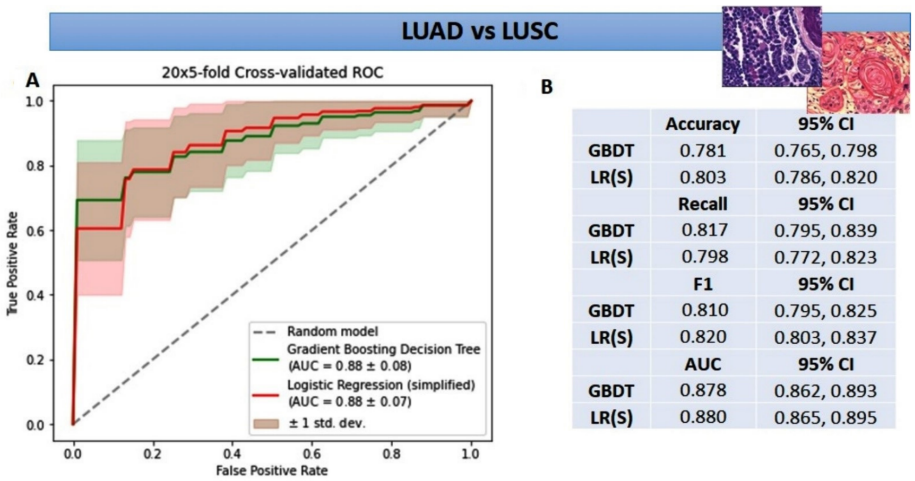
Outline

Interpretable and explainable machine learning





Explanations in medicine

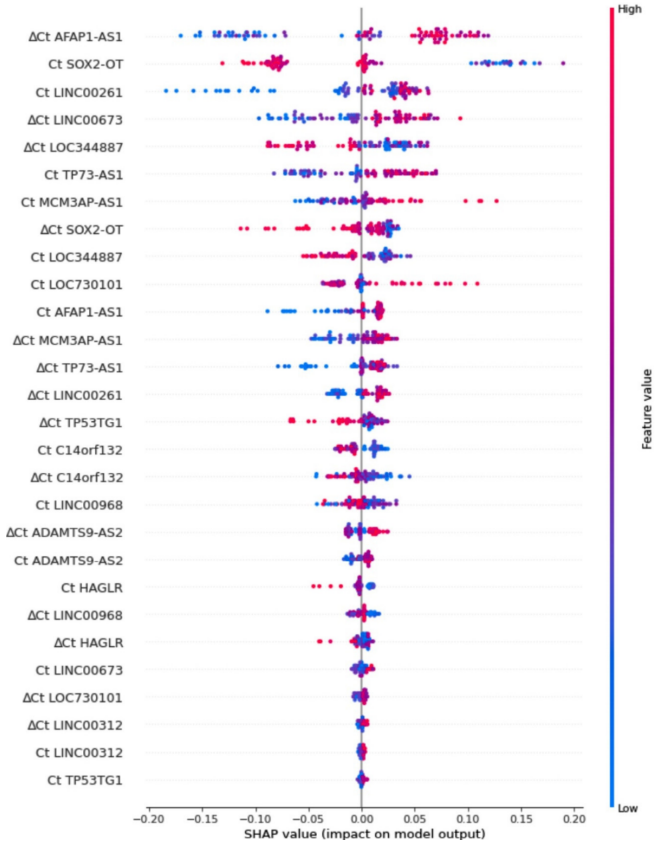


B

	Accuracy	95% CI
GBDT	0.781	0.765, 0.798
LR(S)	0.803	0.786, 0.820
	Recall	**95% CI**
GBDT	0.817	0.795, 0.839
LR(S)	0.798	0.772, 0.823
	F1	**95% CI**
GBDT	0.810	0.795, 0.825
LR(S)	0.820	0.803, 0.837
	AUC	**95% CI**
GBDT	0.878	0.862, 0.893
LR(S)	0.880	0.865, 0.895

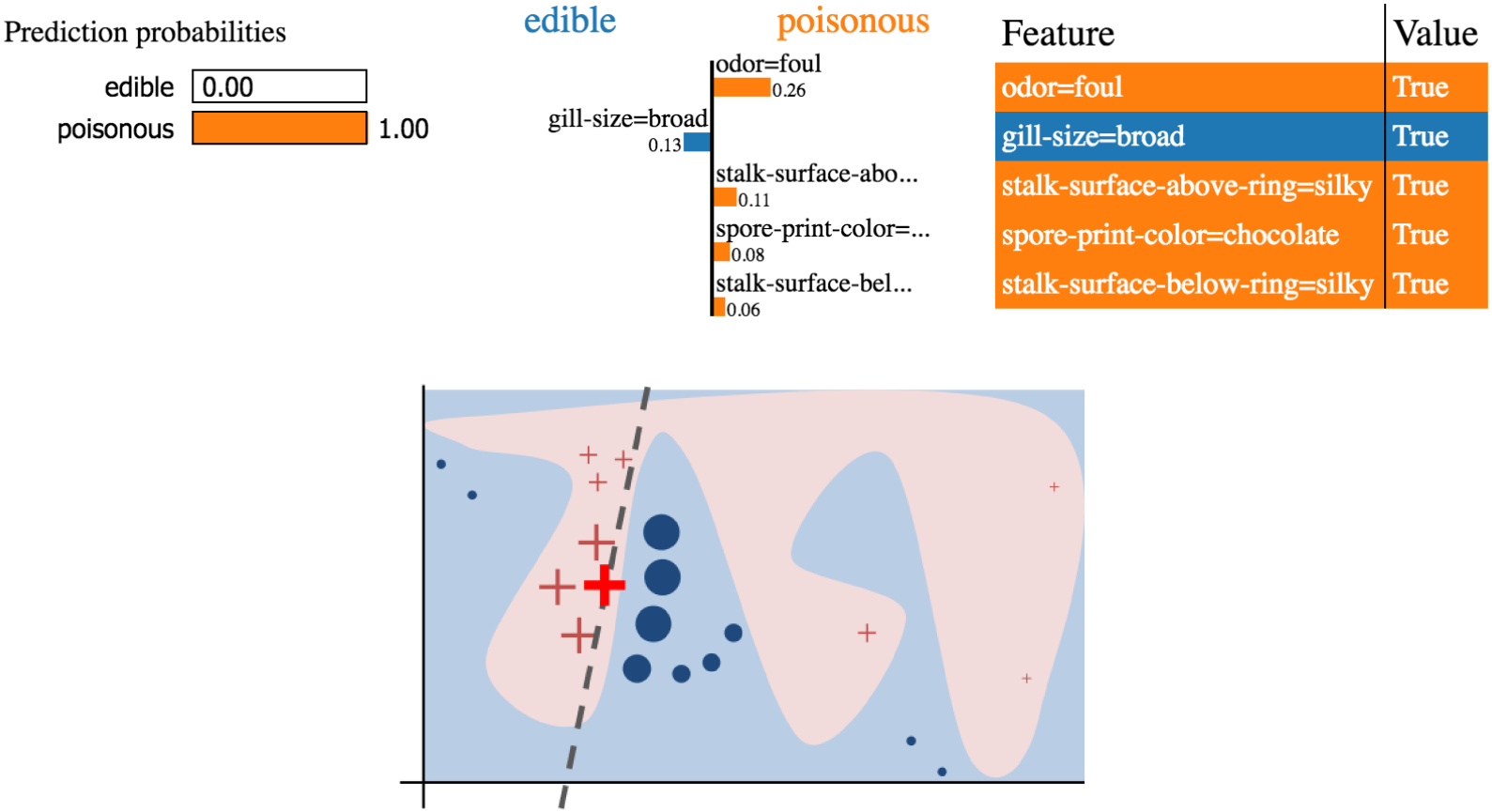
C

	coef	std err	p value	95% CI
dCt AFAP1-AS1	0.4349	0.151	0.004	0.138, 0.731
Ct SOX2-OT	-0.2309	0.067	0.001	-0.362, -0.099
Ct LINC00261	0.2262	0.069	0.001	0.092, 0.361
dCt LINC00673	0.4327	0.206	0.036	0.029, 0.836



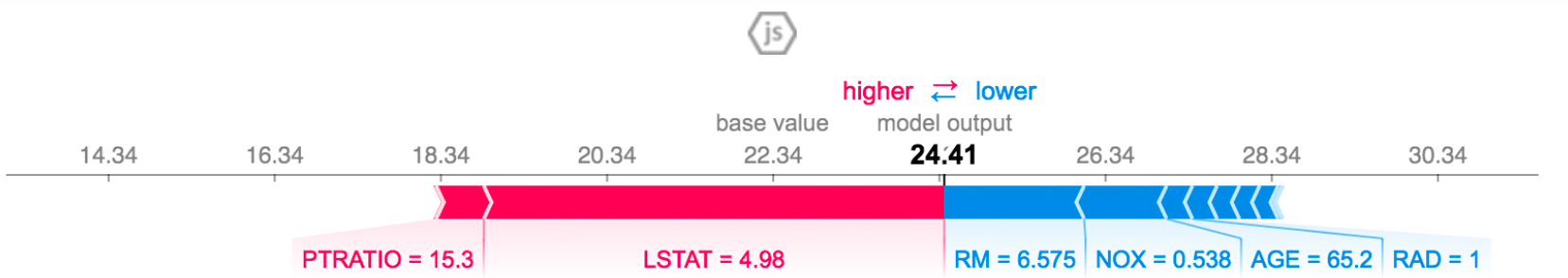
Sulewska et al. [incl. Baniecki & Biecek]. **A Signature of 14 Long Non-Coding RNAs (lncRNAs) as a Step towards Precision Diagnosis for NSCLC**. Cancers, 2022.

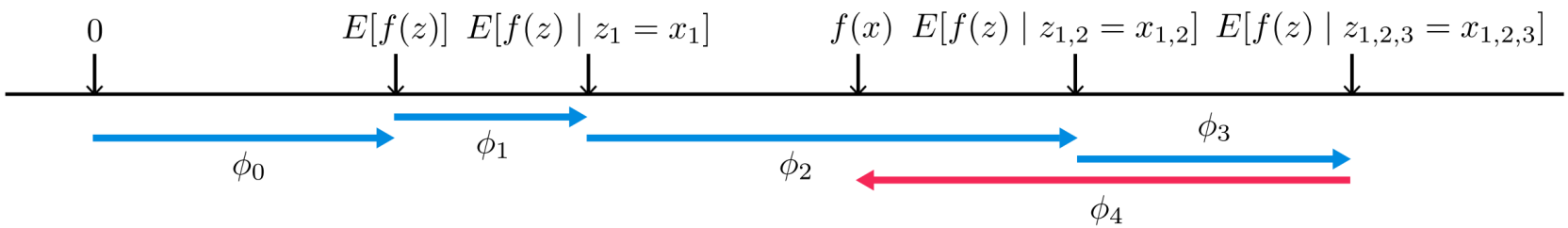
Local Interpretable Model-agnostic Explanations



Ribeiro et al. **"Why Should I Trust You?" Explaining the Predictions of Any Classifier**. KDD, 2016.

SHapley Additive exPlanations





Lundberg & Lee. **A Unified Approach to Interpreting Model Predictions**.

NeurIPS, 2017.

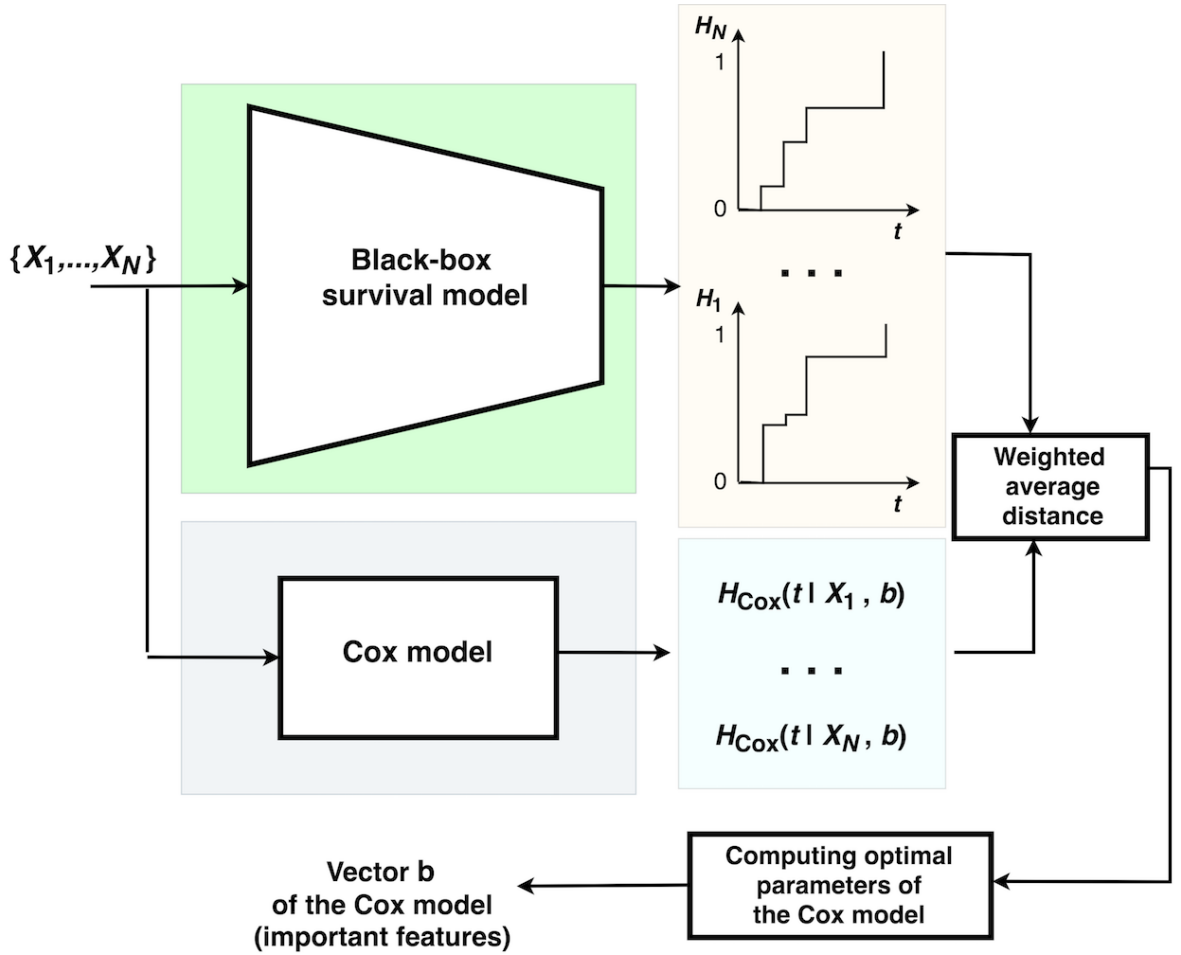
Motivation

- Growing popularity of machine learning and deep learning survival models.
  - **On the one hand:**
    The overoptimistic use of AI models in medicine.
    $\longrightarrow$ The need for a method of validation other than just performance validation.
  - **On the other hand:**
    The complexity and lack of interpretability of AI models hindering their widespread adoption.
    $\longrightarrow$ The need for a method of examining models that would make it possible to understand their operation.
- **Possible solution:** XAI/IML methods suitable for survival analysis task.

Notation

- $\mathbf{x}_i = [x_i^1, x_i^2, \ldots, x_i^p] \in \mathbb{R}^p$ – feature vector, characteristics
- $T_i \in \mathbb{R}$ – survival time
- $C_i \in \mathbb{R}$ – censoring time
- $\delta_i = \mathbb{1}_{T_i \leq C_i}$ – indicator of event of interest's occurrence
- $y_i = \min(T_i, C_i)$ – observed time
- $\mathbb{D} = \{(\mathbf{x}_i, y_i, \delta_i) : i = 1, 2, \ldots, n\}$ – survival dataset
- $t_1 < t_2 < \ldots < t_m$ – distinct times to event of interest from the set $\{y_i : \delta_i = 1; i = 1, 2, \ldots, n\}$
- $S(t)$ – survival function
- $h(t)$ – hazard function
- $H(t)$ – cumulative hazard function

Intuition



Kovalev et al. **SurvLIME: A method for explaining machine learning survival models**. Knowledge Based Systems, 2020.

Theory

Preliminaries

- $T = t_m + \gamma \quad (\gamma > 0)$
- $\Omega = [0, T]$
- $\int_\Omega H(t|\mathbf{x})\mathrm{d}t < \infty$
- Dividing the set Ω into $m+1$ subsets:
  - $\Omega = \bigcup_{j=0,\ldots,m} \Omega_j$
  - $\forall j \in \{0, \ldots, m-1\} \;\; \Omega_j = [t_j, t_{j+1})$, $\Omega_m = [t_m, T]$
  - $\forall j \neq k \;\; \Omega_j \cap \Omega_k = \emptyset$
- $\chi_{\Omega_j}(t) = \begin{cases} 1, & t \in \Omega_j \\ 0, & t \notin \Omega_j \end{cases}$
- $H(t|\mathbf{x}) = \sum_{j=0}^{m} H_j(\mathbf{x}) \cdot \chi_{\Omega_j}(t) \quad (H_j(\mathbf{x}) \geq \epsilon > 0)$

Theory

Formulating optimization problem

- $H_j(\mathbf{x}) \geq \epsilon > 0$
- $\phi(t|\mathbf{x}_k) = \ln\left(H(t|\mathbf{x}_k)\right)$
- $\phi_{\mathsf{Cox}}(t|\mathbf{x}_k, \mathbf{b}) = \ln\left(H_{\mathsf{Cox}}(t|\mathbf{x}_k, \mathbf{b})\right)$

$$
\begin{aligned}
&\phi(t|\mathbf{x}_k) - \phi_{\mathsf{Cox}}(t|\mathbf{x}_k, \mathbf{b}) \\
&= \sum_{j=0}^{m} (\ln H_j(\mathbf{x}_k)) \chi_{\Omega_j}(t) - \sum_{j=0}^{m} \left(\ln(H_{0j} \exp\left(\mathbf{b}^{\mathrm{T}}\mathbf{x}_k\right))\right) \chi_{\Omega_j}(t) \\
&= \sum_{j=0}^{m} \left(\ln H_j(\mathbf{x}_k) - \ln H_{0j} - \mathbf{b}^{\mathrm{T}}\mathbf{x}_k\right) \chi_{\Omega_j}(t) \qquad (1)
\end{aligned}
$$

Theory

Formulating optimization problem

$$
\begin{aligned}
D_{2,k}\left(\phi, \phi_{\mathsf{Cox}}\right) &= \left\|\phi(t|\mathbf{x}_k) - \phi_{\mathsf{Cox}}(t|\mathbf{x}_k, \mathbf{b})\right\|_2^2 \\
&= \int_\Omega \left|\phi(t|\mathbf{x}_k) - \phi_{\mathsf{Cox}}(t|\mathbf{x}_k, \mathbf{b})\right|^2 \mathrm{d}t \\
&= \sum_{j=0}^{m} \left(\ln H_j(\mathbf{x}_k) - \ln H_{0j} - \mathbf{b}^{\mathrm{T}}\mathbf{x}_k\right)^2 \left[\int_\Omega \chi_{\Omega_j}(t)\right] \\
&= \sum_{j=0}^{m} \left(\ln H_j(\mathbf{x}_k) - \ln H_{0j} - \mathbf{b}^{\mathrm{T}}\mathbf{x}_k\right)^2 (t_{j+1} - t_j)
\end{aligned}
\tag{2}
$$

Theory

Formulating optimization problem

- **weights of the examples:** $w_k = K(\mathbf{x}, \mathbf{x}_k)$, where $K(\cdot, \cdot)$ is a kernel
- **weights straightening functions** $\phi(t|\mathbf{x})$ **and** $\phi_{\textbf{Cox}}(t|\mathbf{x}, \mathbf{b})$: $v(t|\mathbf{x}) = \frac{H(t|\mathbf{x})}{\phi(t|\mathbf{x})} = \frac{H(t|\mathbf{x})}{\ln(H(t|\mathbf{x}))}$

Optimization problem

$$\min_{\mathbf{b}} \sum_{k=1}^{N} w_k \sum_{j=0}^{m} v_{kj}^2 \left(\ln H_j(\mathbf{x}_k) - \ln H_{0j} - \mathbf{b}^{\mathrm{T}}\mathbf{x}_k\right)^2 (t_{j+1} - t_j) \quad (3)$$

Algorithm

Algorithm The algorithm for computing vector $\mathbf{b}$ for point $\mathbf{x}$

Require: Training set D; point of interest $\mathbf{x}$; the number of generated points N; the black-box survival model for explaining $f(\mathbf{x})$

Ensure: Vector $\mathbf{b}$ of important features

1: Compute the baseline cumulative hazard function $H_0(t)$ of the approximating Cox model on dataset D by using the Nelson–Aalen estimator
2: Generate $N-1$ random nearest points $\mathbf{x}_k$ in a local area around $\mathbf{x}$, point $\mathbf{x}$ is the N-th point
3: Get the prediction of the cumulative hazard function $H(t|\mathbf{x}_k)$ by using the black-box survival model
4: Compute weights of the generated points from neighbourhood
5: Compute weights for *straightening* logarithm functions
6: Find vector $\mathbf{b}$ by solving the convex optimization problem 3

Example: SurvLIME of CPH

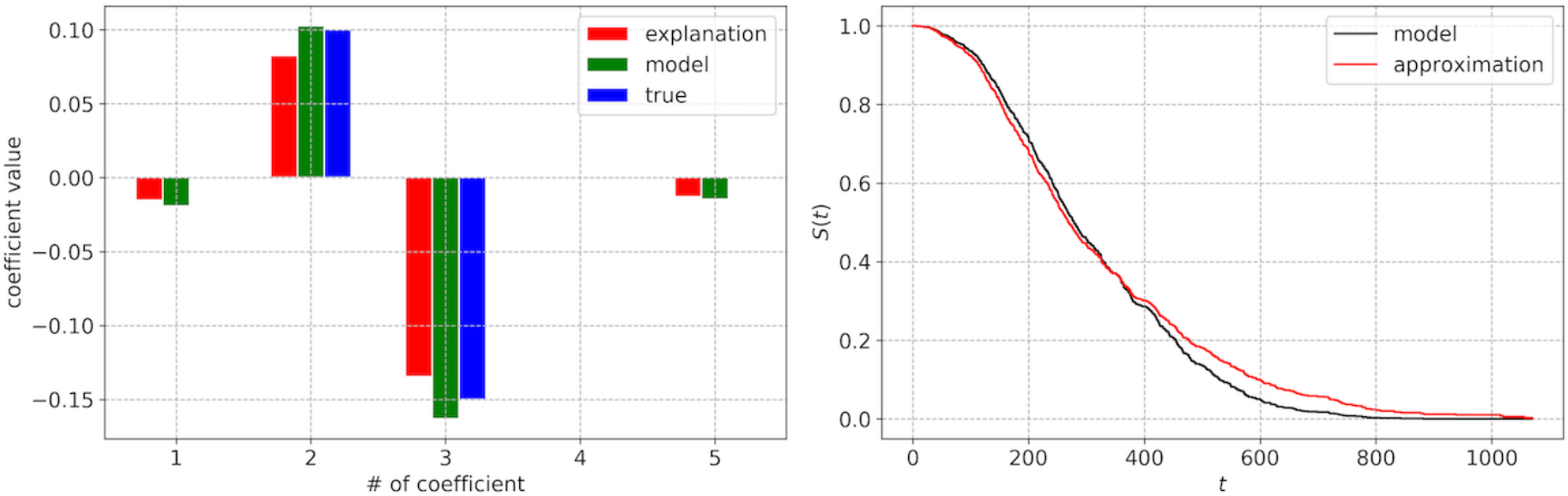


Figure: Example of SurvLIME for one observation. Comparison with coefficients of the Cox model trained on a synthetic dataset.

Kovalev et al. **SurvLIME: A method for explaining machine learning survival models**. Knowledge Based Systems, 2020.

Example: SurvLIME of RSF

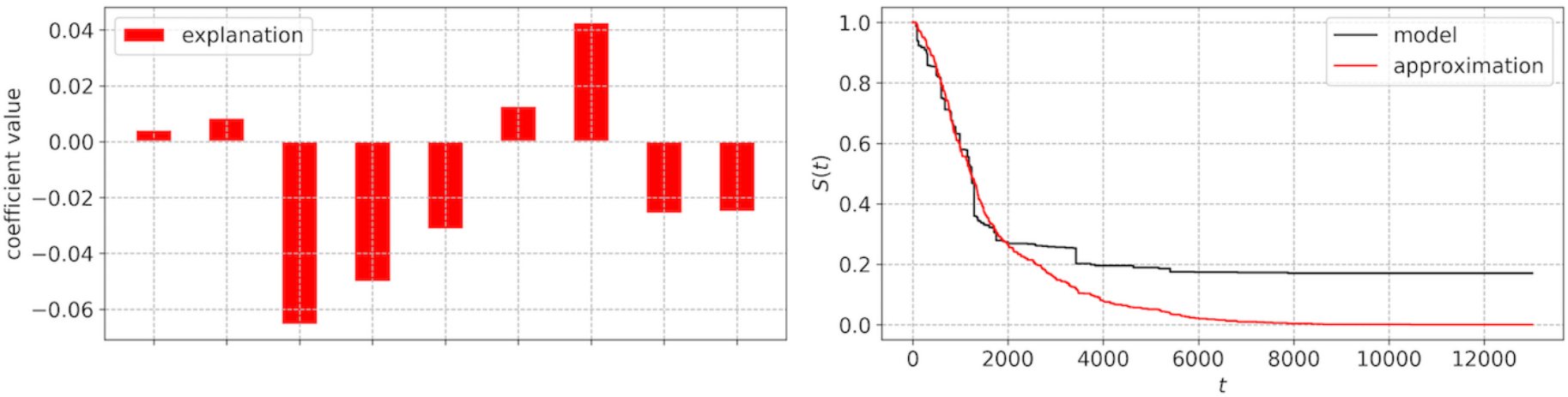


Figure: Example of SurvLIME for a random survial forest trained on the CML dataset.

Kovalev et al. **SurvLIME: A method for explaining machine learning survival models**. Knowledge Based Systems, 2020.

Other related methods

- SurvLIME-KS (Kolmogorov-Smirnov bounds) (Kovalev & Utkin, 2020)
- UncSurvEx (prediction uncertainty) (Utkin et al., 2021)
- Counterfactual explanations (Kovalev et al., 2021)

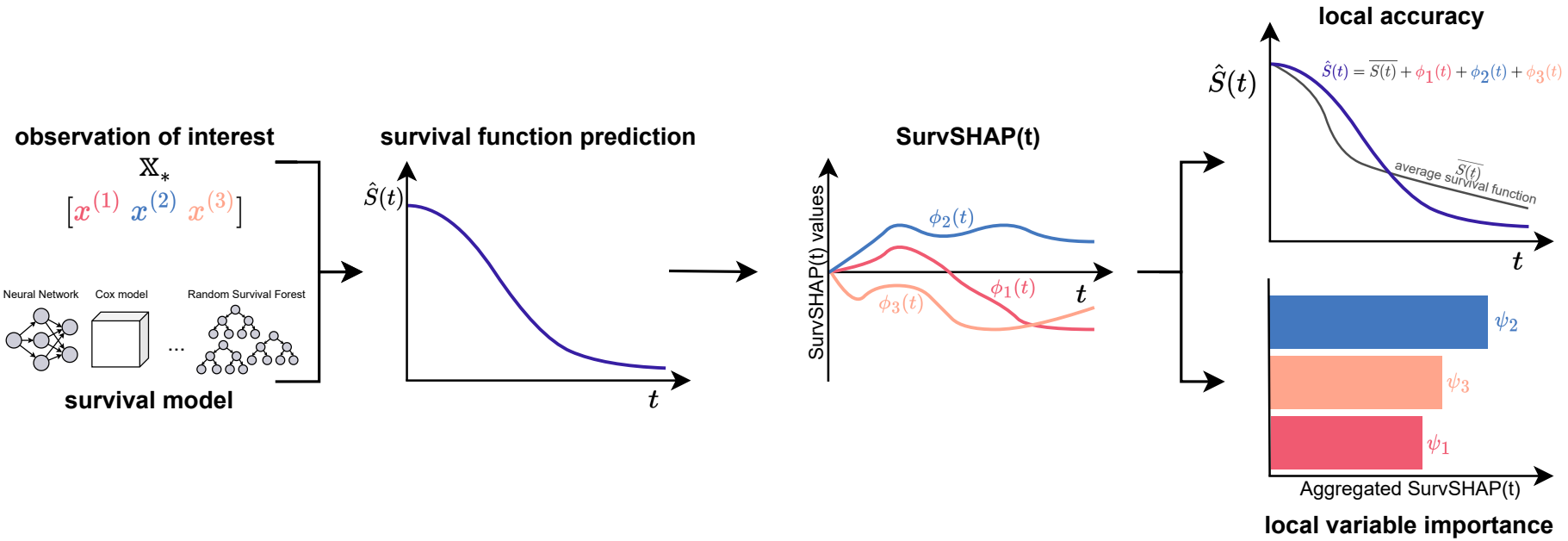
Limitations & observations

- The lack of publicly available open-source implementations of the methods described in the literature.
- Ambiguous description of optimization.
- Problem with reproducibility – lack of availability of data generated for experiments and their codes.
- None of the known approaches supply **time-dependent** explanations.

Time-dependent explanations

- Including the time dimension, crucial for predicting the survival conditional probability distribution, in the final explanation.
- Providing insights into how each variable influences the model's response (survival function) at each time point.
- Especially useful for survival models that can model complex dependencies (e.g., random survival forest (Ishwaran et al., 2008)).

Intuition



Krzyziński, Spytek, Baniecki, Biecek. **SurvSHAP(t): Time-dependent explanations of machine learning survival models**. Preprint in review, 2022.

Theory

Goal: For observation of interest $\mathbf{x}_*$, to assign an attribution $\phi_t(\mathbf{x}_*, d)$ for the prediction:

- to the value of each variable $x^{(d)}$, $d \in \{1, 2, \ldots, p\}$, included in the model,
- at any selected time point t.

In this way, the **SurvSHAP(t)** functions $[\phi_{t_1}(\mathbf{x}_*, d), \phi_{t_2}(\mathbf{x}_*, d), \ldots, \phi_{t_m}(\mathbf{x}_*, d)]$ are generated.

Theory

SurvSHAP(t) values:

$$\phi_t(\mathbf{x}_*, d) = \frac{1}{|\Pi|} \sum_{\pi \in \Pi} e_{t,\mathbf{x}_*}^{\text{before}(\pi,d)\cup\{d\}} - e_{t,\mathbf{x}_*}^{\text{before}(\pi,d)}, \tag{4}$$

where Π is a set of all permutations of p variables and $\text{before}(\pi, d)$ denotes a subset of predictors that are before d in the ordering $\pi \in \Pi$.

Theory

Normalized SurvSHAP(t) values:

$$\phi_t^*(\mathbf{x}_*, d) = \frac{\phi_t(\mathbf{x}_*, d)}{\sum_{j=1}^{p} |\phi_t(\mathbf{x}_*, j)|}. \tag{5}$$

Local variable importance based on SurvSHAP(t):

$$\psi(\mathbf{x}, d) = \int_0^{t_m} |\phi_t(\mathbf{x}, d)| \, \mathrm{d}t. \tag{6}$$

Evaluation metrics

- **Local accuracy** (time-dependent adaptation of a local accuracy metric proposed by Lundberg et al. (2020)):

$$\sigma(t) = \sqrt{\frac{\mathbb{E}(\hat{S}(t, \mathbf{x}) - \sum_i \phi_t(\mathbf{x}, i))^2}{\mathbb{E}\hat{S}(t, \mathbf{x})^2}}. \tag{7}$$

- **Changing Sign Proportion (CSP)**:

$$CSP_{\alpha, t_\mathrm{s}, t_\mathrm{e}} = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1} \left(\left| \bigcup_{t \in [t_\mathrm{s}, t_\mathrm{e}]} \{t : \phi_t(\mathbf{x}_i, d) \geq 0\} \right| > \alpha \cdot |[t_\mathrm{s}, t_\mathrm{e}]| \right) \mathbb{1} \left(\left| \bigcup_{t \in [t_\mathrm{s}, t_\mathrm{e}]} \{t : \phi_t(\mathbf{x}_i, d) \leq 0\} \right| > \alpha \cdot |[t_\mathrm{s}, t_\mathrm{e}]| \right). \tag{8}$$

Evaluation metrics

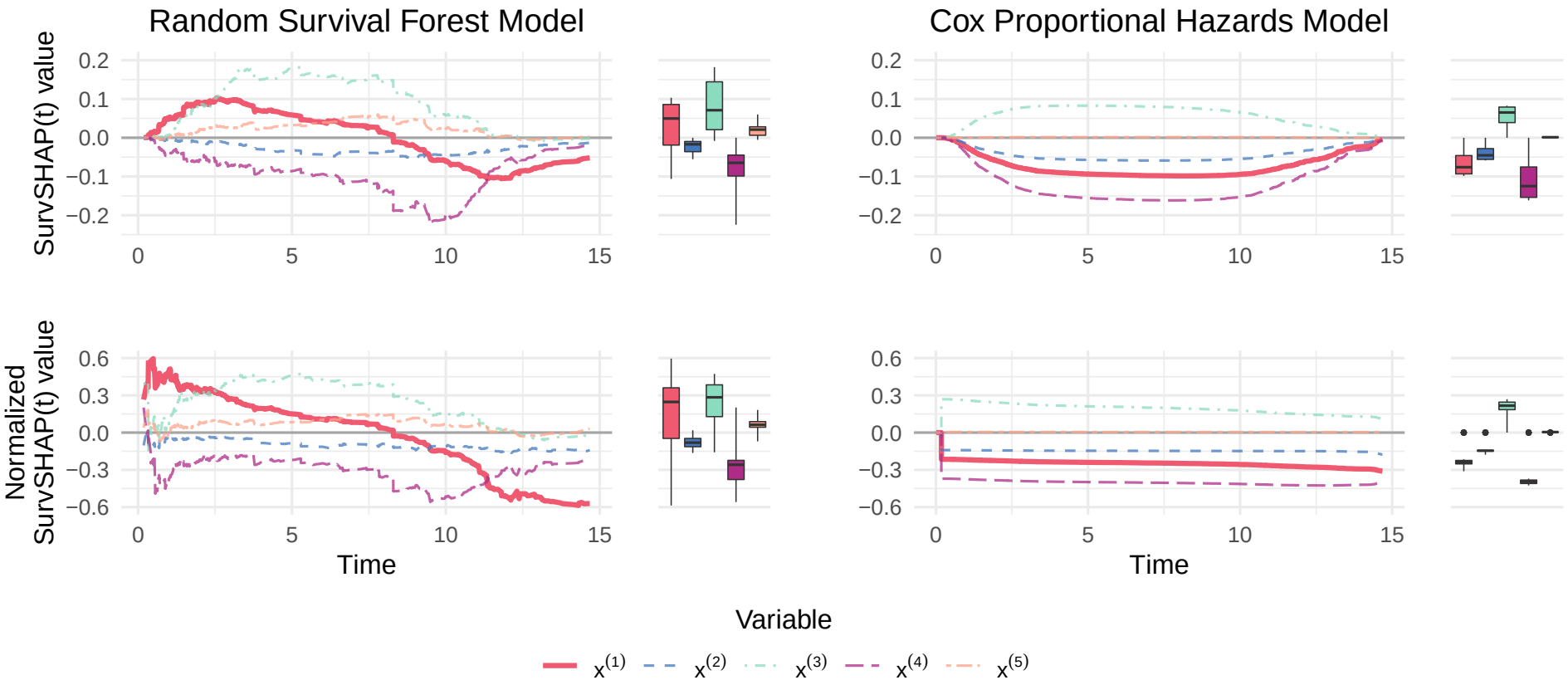
- **Additive hyperbolic Kendall's τ_h rank correlation coefficient** [Vigna (2015)].
- **GT-Shapley** [Lundberg et al. (2020)]:

$$\rho(t) = \frac{1}{n}\sum_{i=1}^{n} \texttt{Pearson}([\phi_t(\mathbf{x}_i, d)]_{1\leq d\leq p}, [\phi_t^{true}(\mathbf{x}_i, d)]_{1\leq d\leq p}), \tag{9}$$

- **Normalized RMSE**:

$$\text{normalized RMSE}(t, d) = \sqrt{\frac{\mathbb{E}(\phi_t(\mathbf{x}, d) - \phi_t^{true}(\mathbf{x}, d))^2}{\mathbb{E}\phi_t^{true}(\mathbf{x}, d)^2}}. \tag{10}$$

Synthetic data: capturing a time-dependent effect



$$h(t) = h_0(t) \cdot \exp[(-0.9 + 0.1\mathbf{t} + 0.9\ln(\mathbf{t}))\, x^{(1)} \ldots]$$

SurvSHAP(t) better approximates feature attributions

Table: Average τ_h **correlations** of the variable importance rankings according to explanations against the ground-truth ranking in the CPH model (**higher** is better).

	SurvLIME	**SurvSHAP(t)**
`dataset0`	0.763	0.917
`dataset1`	0.454	0.745

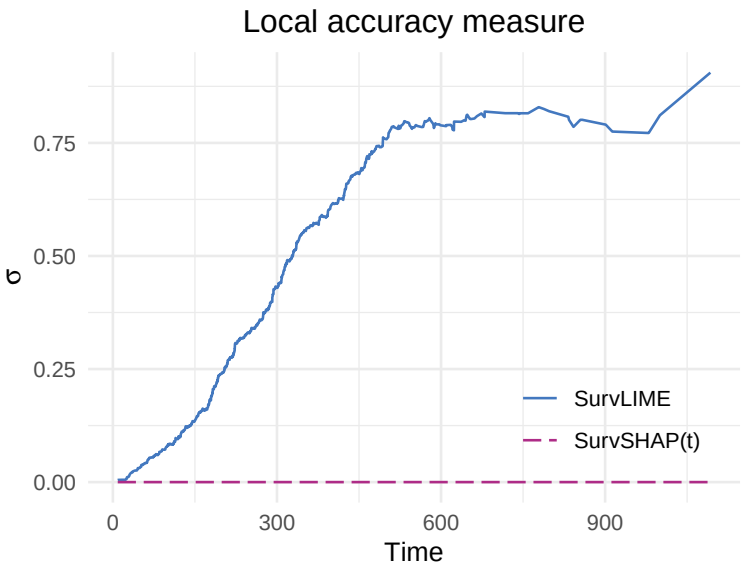


Figure: Normalized standard deviation of the difference between the RSF model output and the explanation (**lower** is better). **Note:** the curve for SurvSHAP(t) coincides with the x-axis.

Real-world use case

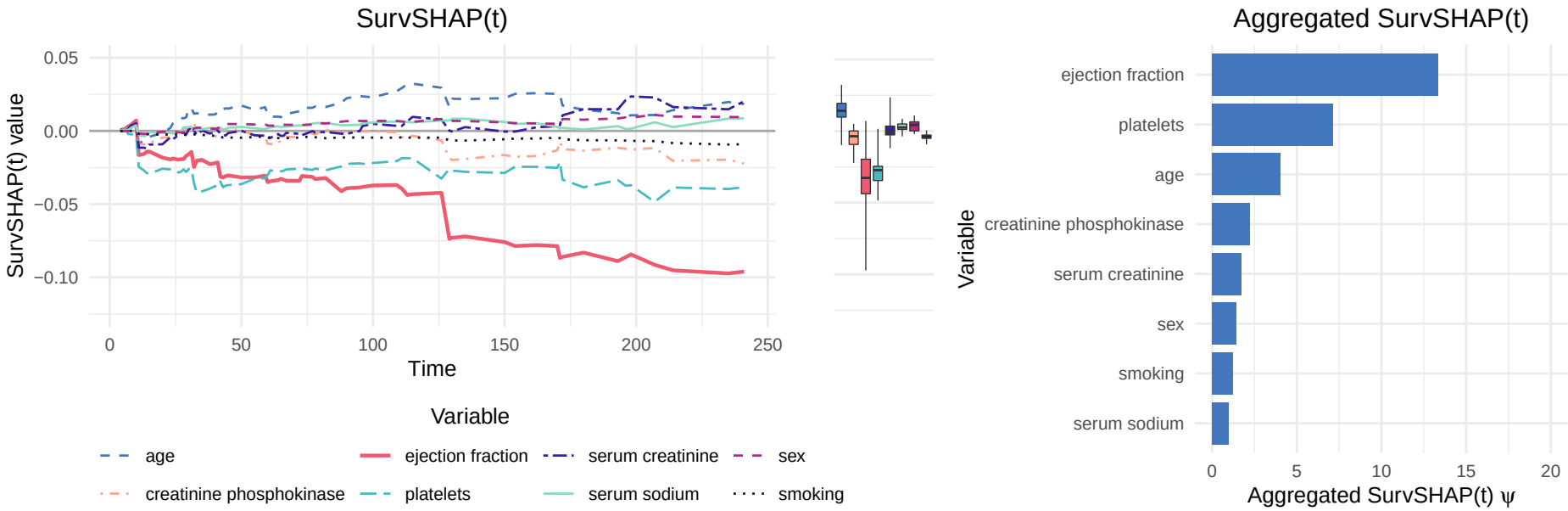


Figure: Explanation results for the selected observation and RSF model trained on the `heart_failure` dataset: SurvSHAP(t) values (**left**) and aggregated SurvSHAP(t) values – variable importance measure (**right**).

Comparison to SurvLIME

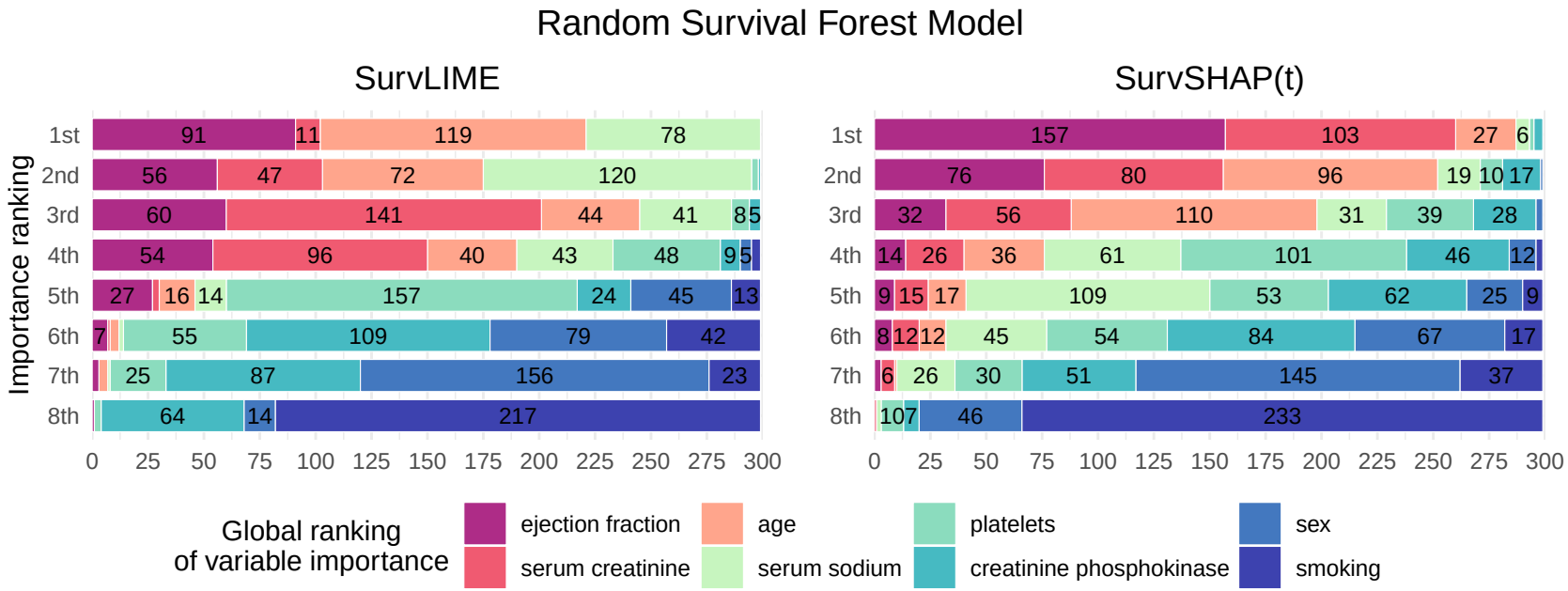
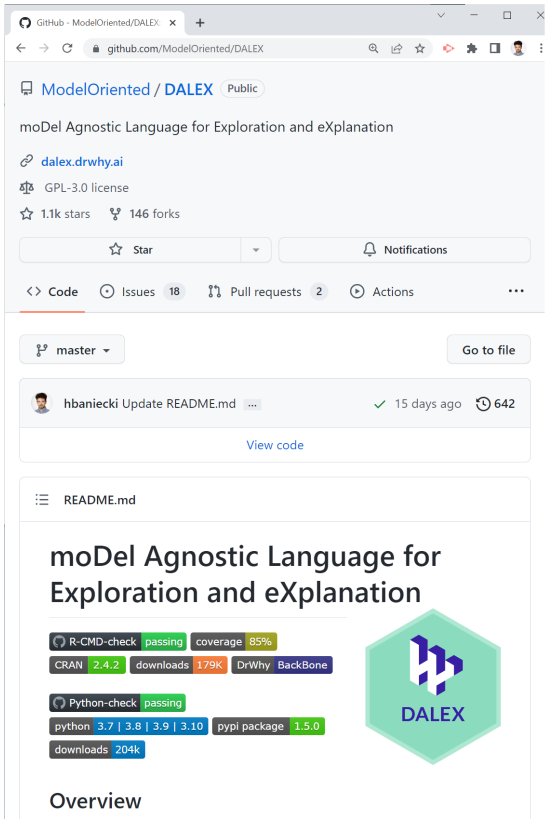
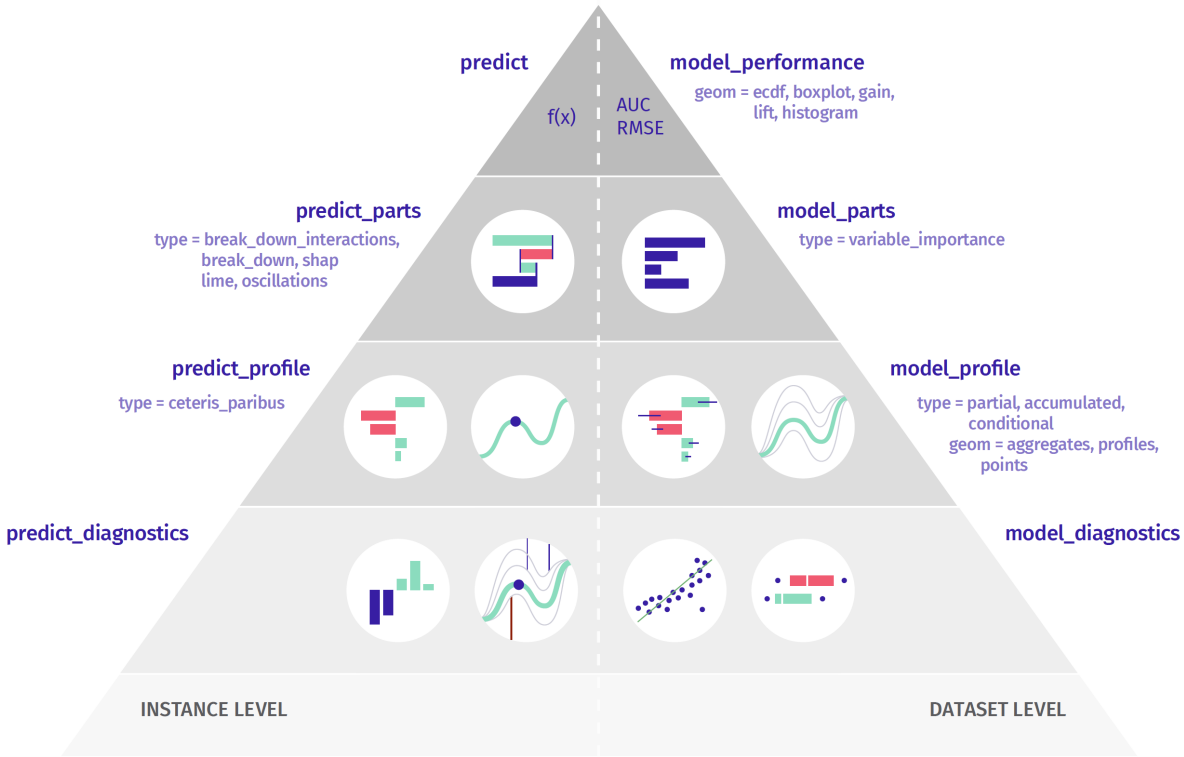


Figure: Juxtaposition of local and global importance rankings for predictions of the RSF model. The colors are specifically sorted from purple to blue showing the global ranking of variables in each model.

DALEX: Explainable machine learning in R





survex functionalities

survex R package:

- Extensions of methods from (Molnar, 2020; Biecek & Burzykowski, 2021) and survival-specific methods (Kovalev et al., 2020; Krzyziński et al., 2022).
- `explain()` interface based on DALEX.
- other useful utilities – transforming output types, visualizations etc.

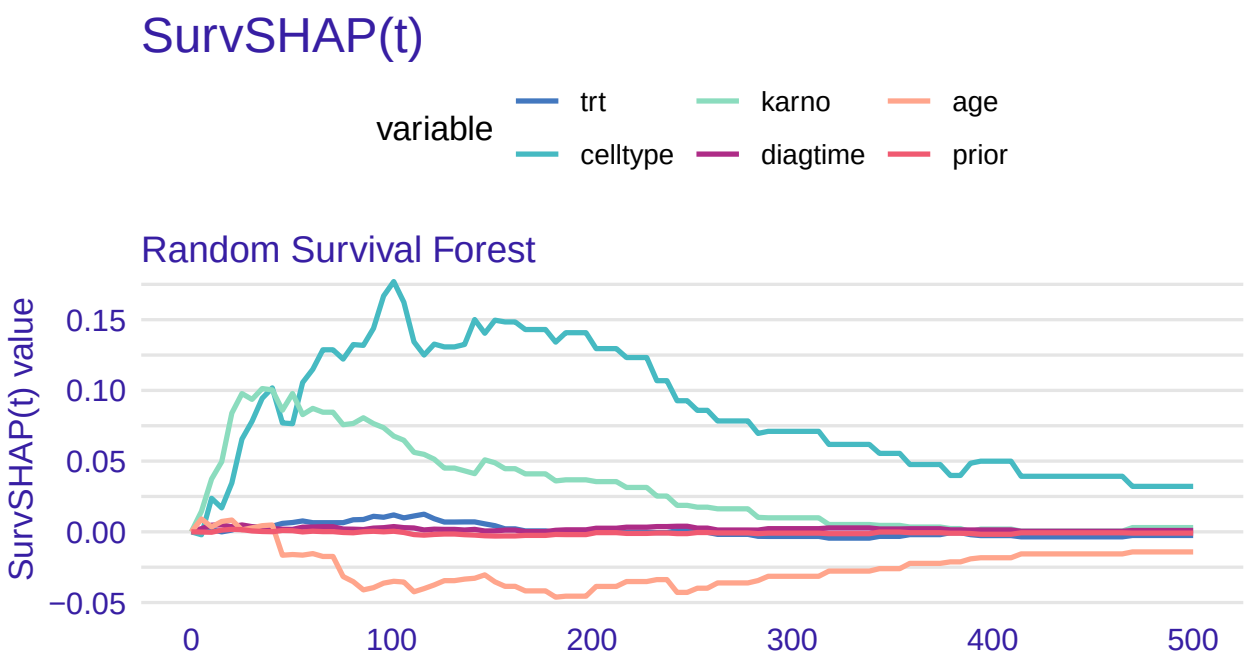
`https://github.com/modeloriented/survex`

Unified prediciton interface

- Models from different R packages, use different functions to make predictions.
- survex unifies this by overloading the `predict()` function for explainers.
- It allows for predicting different types of output – survival function, cumulative hazard function and risk scores.

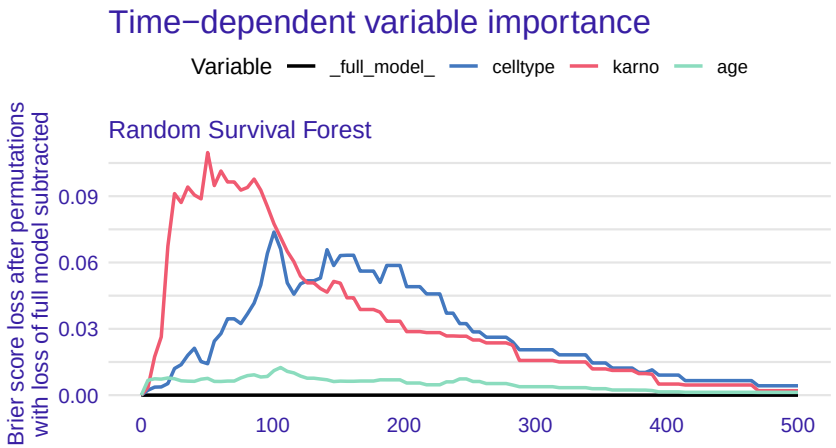
```
# Making model predictions
predict(explainer, data = df, times=1:100, output_type="survival")
predict(explainer, data = df, times=1:100, output_type="chf")
```

SurvSHAP(t)



$$\phi_t(\mathbf{x}_*, d) = \frac{1}{|\Pi|} \sum_{\pi \in \Pi} e_{t,\mathbf{x}_*}^{\text{before}(\pi,d)\cup\{d\}} - e_{t,\mathbf{x}_*}^{\text{before}(\pi,d)} \tag{11}$$

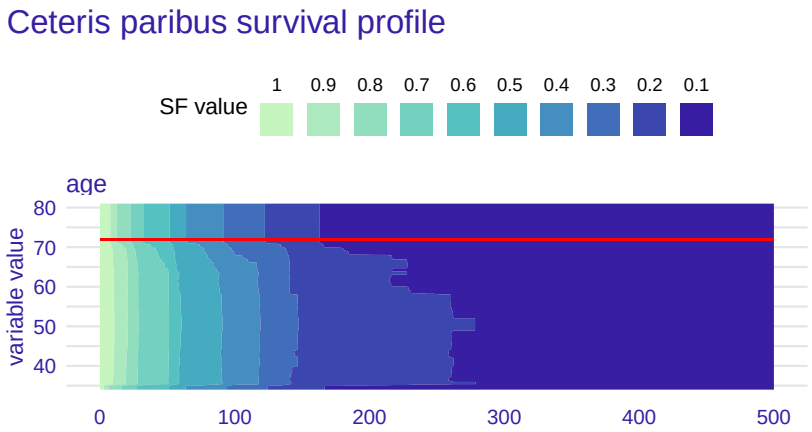
Permutational variable importance



$$v^j(t) = \frac{1}{B}\sum_{i=1}^{B} \mathcal{L}(y, f(X^{*j}, t)) - \mathcal{L}(y, f(X, t)) \quad (12)$$

B – number of considered permutations, $\mathcal{L}(y, f(X, t))$ – loss function, X^{*j} – model input with the j-th column permuted.

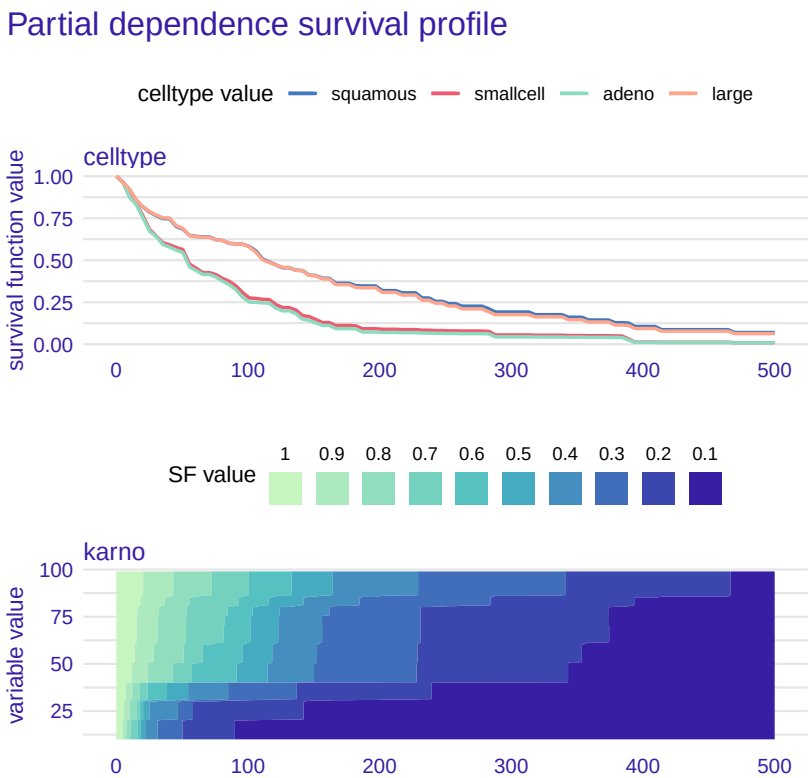
Ceteris paribus



$$h^{f,j}_{\mathbf{x}_*}(z,t) = f(\mathbf{x}_*^{j|=z}, t) \qquad (13)$$

$x_i^{j|=z}$ – i-th observation with the j-th column set to the value z

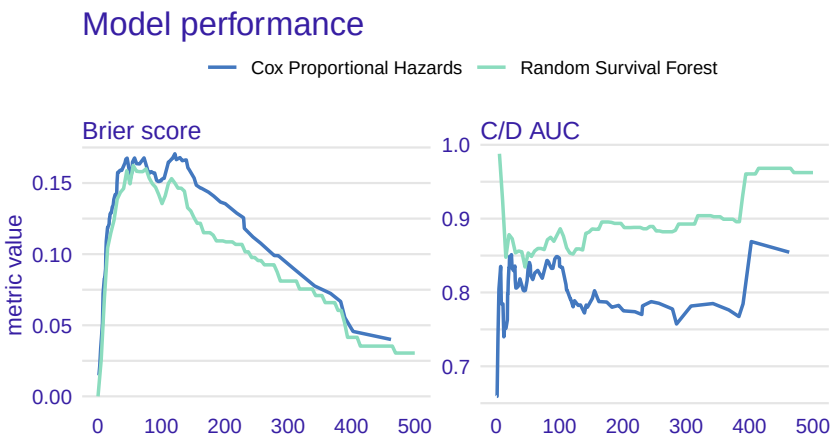
Partial dependence



$$\hat{g}_{PD}(z,t) = \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x}_i^{j|=z}, t) \tag{14}$$

n – number of samples,
$x_i^{j|=z}$ – i-th observation with the j-th column set to the value z

Measuring model performance



- Implementation of both time-dependent and single value metrics.
- C/D AUC, Brier score
- Integrated C/D AUC, Integrated Brier score, Harrell's Concordance index

survex + mlr3proba

- vignette
- basic example

```
library(survex); library(mlr3proba); library(mlr3learners)
library(mlr3extralearners); library(survival)

task <- as_task_surv(veteran,
                     id="right_censored",
                     time = "time",
                     event = "status")
learner <- lrn("surv.parametric")
learner$train(task)
ex <- explain(learner,
              data = veteran[, -c(3,4)],
              y = Surv(veteran$time, veteran$status))
model_performance(ex) |> plot()
model_profile(ex) |> plot()
```

Collaboration with physicians

- Donizy et al. [incl. Krzyziński & Biecek]. **Machine learning models demonstrate that clinicopathologic variables are comparable to gene expression prognostic signature in predicting survival in uveal melanoma**. European Journal of Cancer, 2022.
- **Next step:** Using explanations of survival models for selection of additional features and performance improvement.

Other ideas

- Checking the trustworthiness of machine learning survival models with use of the survex package:
  - intro: use-case for uveal melanoma TCGA dataset;
  - further plan: creating benchmark using more datasets, constitute trustworthiness checklist.

References

Przemyslaw Biecek and Tomasz Burzykowski. *Explanatory Model Analysis*. Chapman and Hall/CRC, 2021.

Hemant Ishwaran, Udaya B. Kogalur, Eugene H. Blackstone, and Michael S. Lauer. Random survival forests. *The Annals of Applied Statistics*, 2(3): 841–860, 2008. doi: 10.1214/08-AOAS169.

Maxim S. Kovalev and Lev V. Utkin. A robust algorithm for explaining unreliable machine learning survival models using the Kolmogorov–Smirnov bounds. *Neural Networks*, 132:1–18, 2020. doi: 10.1016/j.neunet.2020.08.007.

Maxim S. Kovalev, Lev V. Utkin, and Ernest M. Kasimov. SurvLIME: A method for explaining machine learning survival models. *Knowledge-Based Systems*, 203:106164, 2020. doi: 10.1016/j.knosys.2020.106164.

Maxim S. Kovalev, Lev V. Utkin, Frank Coolen, and Andrei V. Konstantinov. Counterfactual Explanation of Machine Learning Survival Models. *Informatica*, 32(4):817–847, 2021. doi: 10.15388/21-INFOR468.

Mateusz Krzyziński, Mikołaj Spytek, Hubert Baniecki, and Przemysław Biecek. Survshap(t): Time-dependent explanations of machine learning survival models. *arXiv preprint arXiv:2208.11080*, 2022.

Scott M Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence*, 2(1):56–67, 2020. doi: 10.1038/s42256-019-0138-9.

Christoph Molnar. *Interpretable Machine Learning*. 2020.

Lev V. Utkin, Vladimir S. Zaborovsky, Maxim S. Kovalev, Andrei V. Konstantinov, Natalia A. Politaeva, and Alexey A. Lukashin. Uncertainty Interpretation of the Machine Learning Survival Model Predictions. *IEEE Access*, 9:120158–120175, 2021. doi: 10.1109/ACCESS.2021.3108341.

Sebastiano Vigna. A Weighted Correlation Index for Rankings with Ties. In *International Conference on World Wide Web (WWW)*, pp. 1166–1176, 2015. ISBN 9781450334693. doi: 10.1145/2736277.2741088.