Swiss Re

Privacy Preserving Machine Learning

SAV Après-Midi, 31 August 2023

Daniel Meier, Swiss Re
Juan Ramón Troncoso-Pastoriza, Tune Insight

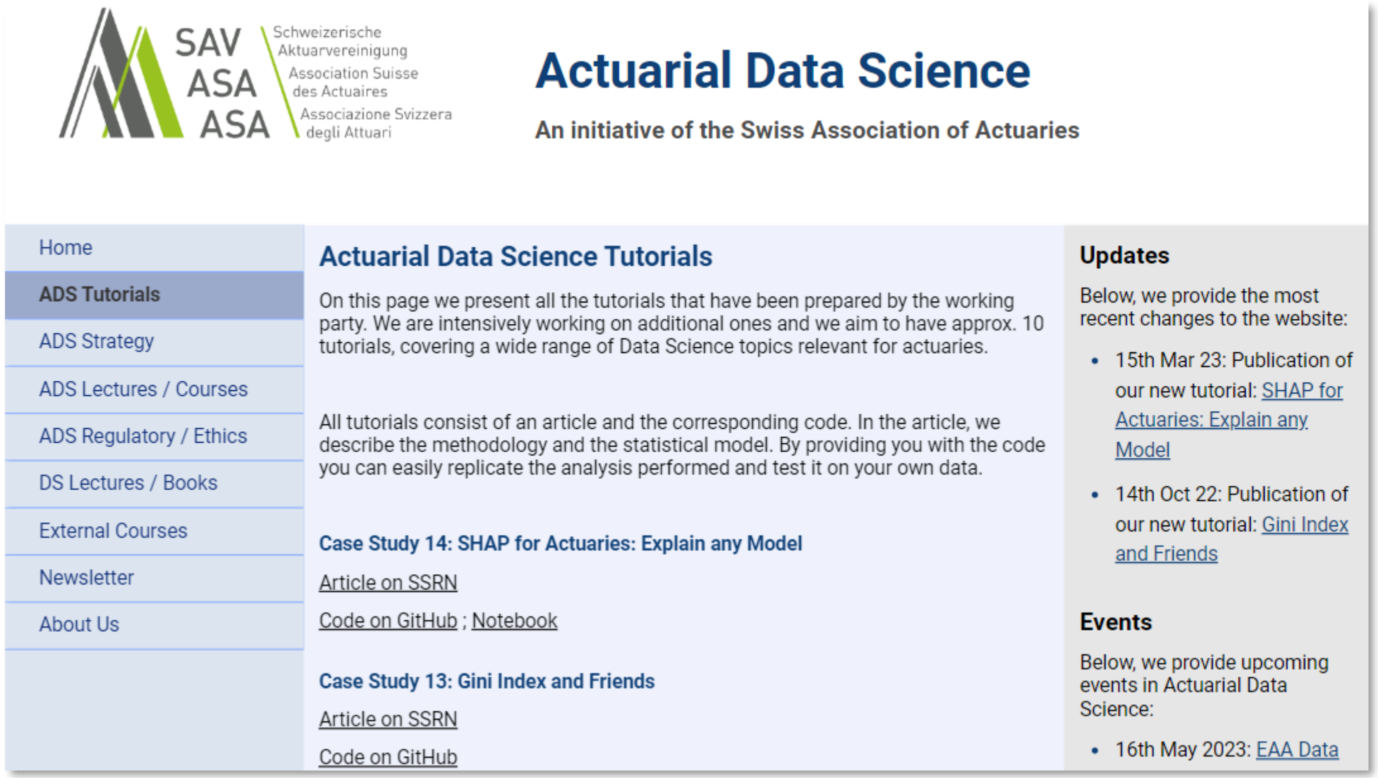
Key take-aways

1. Limited data availability for sensitive personal (Life & Health) data in practice (e.g., nFADP, 1 September 2023)
2. For sufficiently large and dense datasets, ML/DL methods outperform traditional models, creating value for policyholders and insurance companies
3. Privacy preserving methods can help to access data

Agenda

- Creation of synthetic health datasets
- Introducing 3 models to create health risk scores: Logistic regression, Cox regression, neural networks
- Homomorphic encryption

Paper and code soon available at actuarialdatascience.org

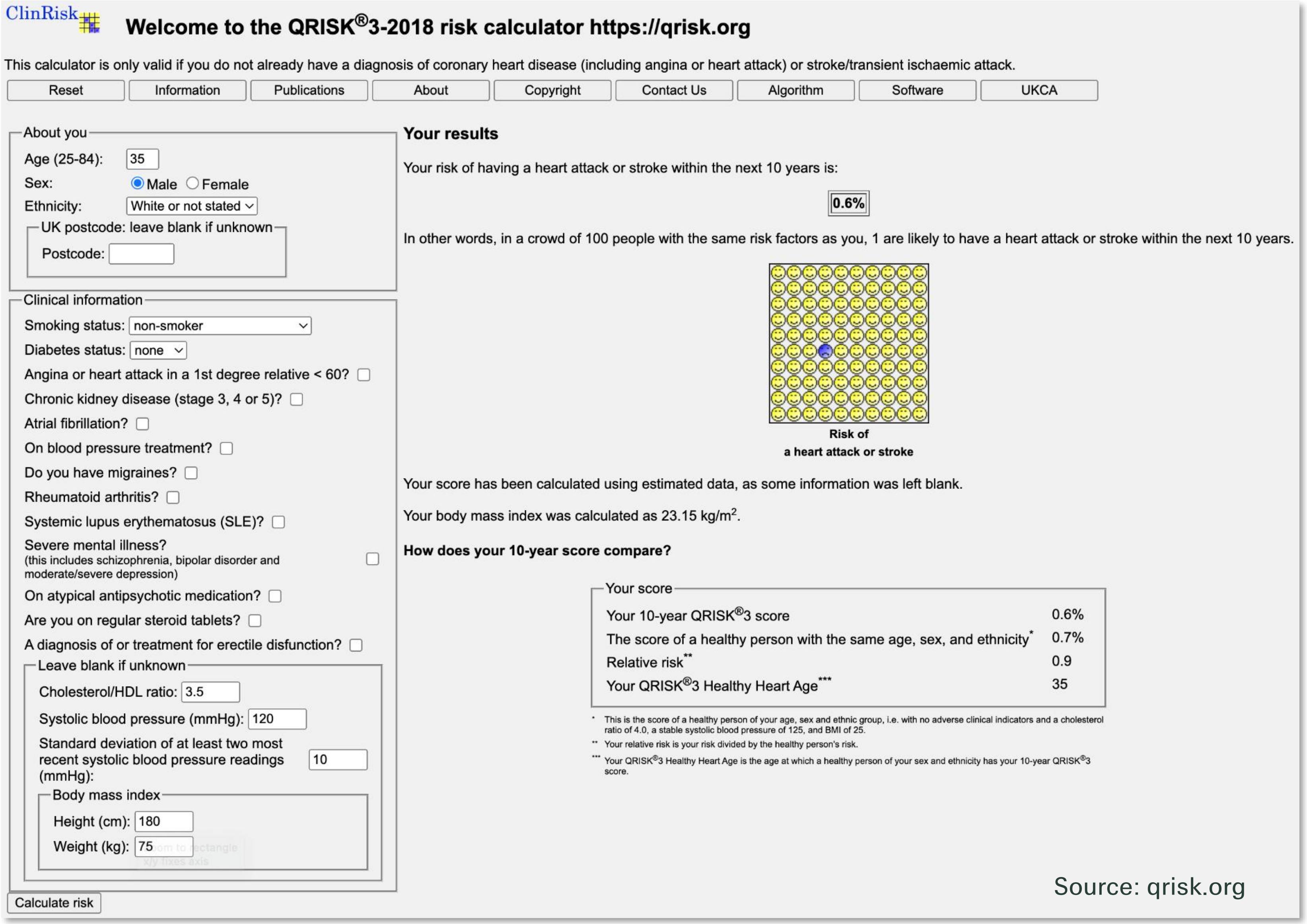


(Publicly) available health datasets

- CPRD, https://cprd.com/data
- MIMIC, https://physionet.org/about/database/
- IPUMS, https://healthsurveys.ipums.org/
- NHANES, https://www.cdc.gov/nchs/nhanes/
- Nightingale, https://docs.nightingalescience.org/
- UK Biobank, https://www.ukbiobank.ac.uk/
- IHME, https://ghdx.healthdata.org/
- ...
- See also longitudinal study for other health datasets

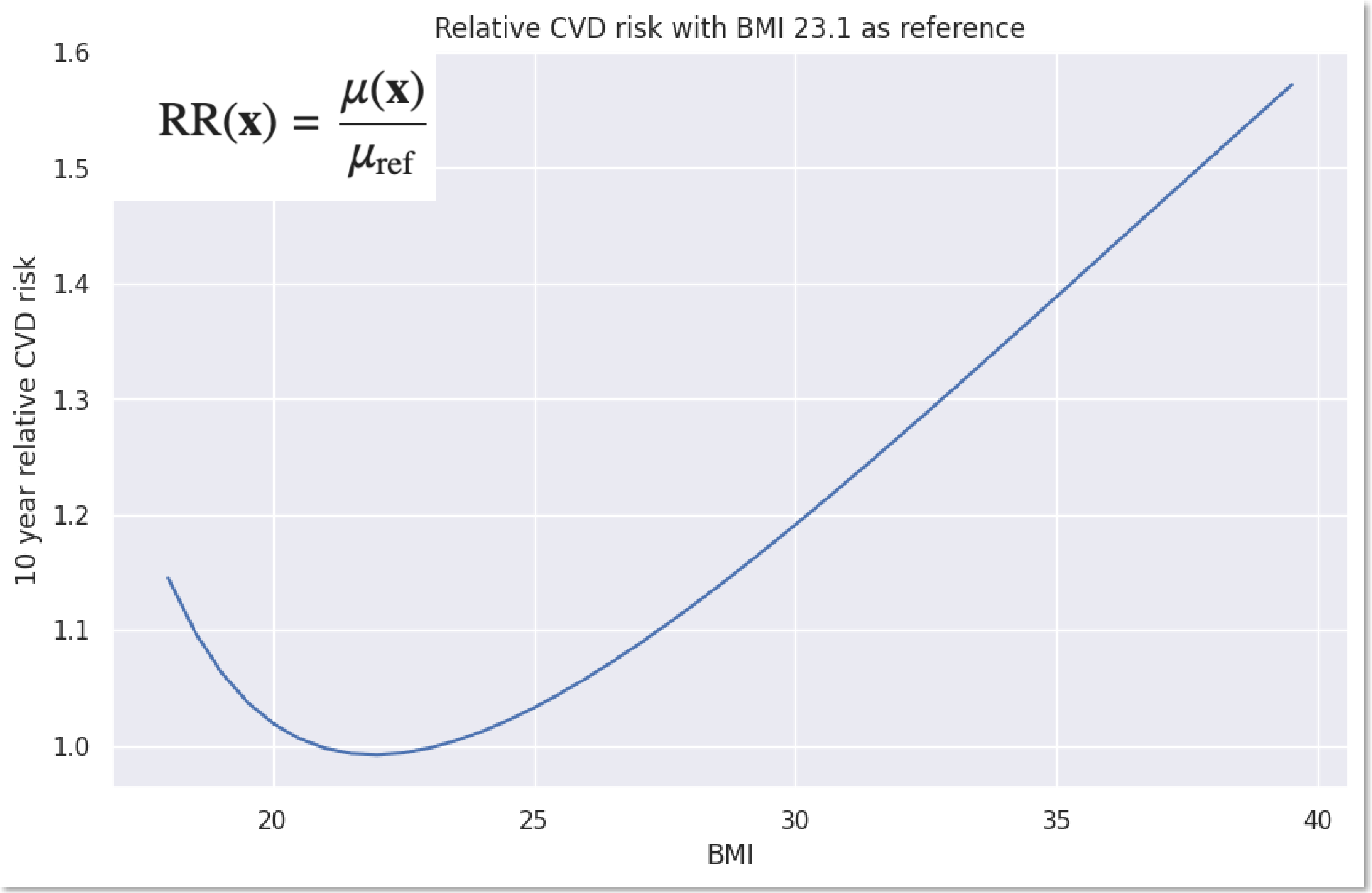
- Often, access is restricted to academic institutions and/or limited to a pre-defined research topic
- Data volumes (and density) rather too low for ML
- Access to more data sources (e.g., hospitals, GPs, insurance companies, etc.) – in a privacy preserving manner – is needed

Health risk scores, e.g., QRISK3 providing 10-year risk of a cardio-vascular disease (CVD)



Source: qrisk.org

Relative risk with respect to a reference person of same age, gender

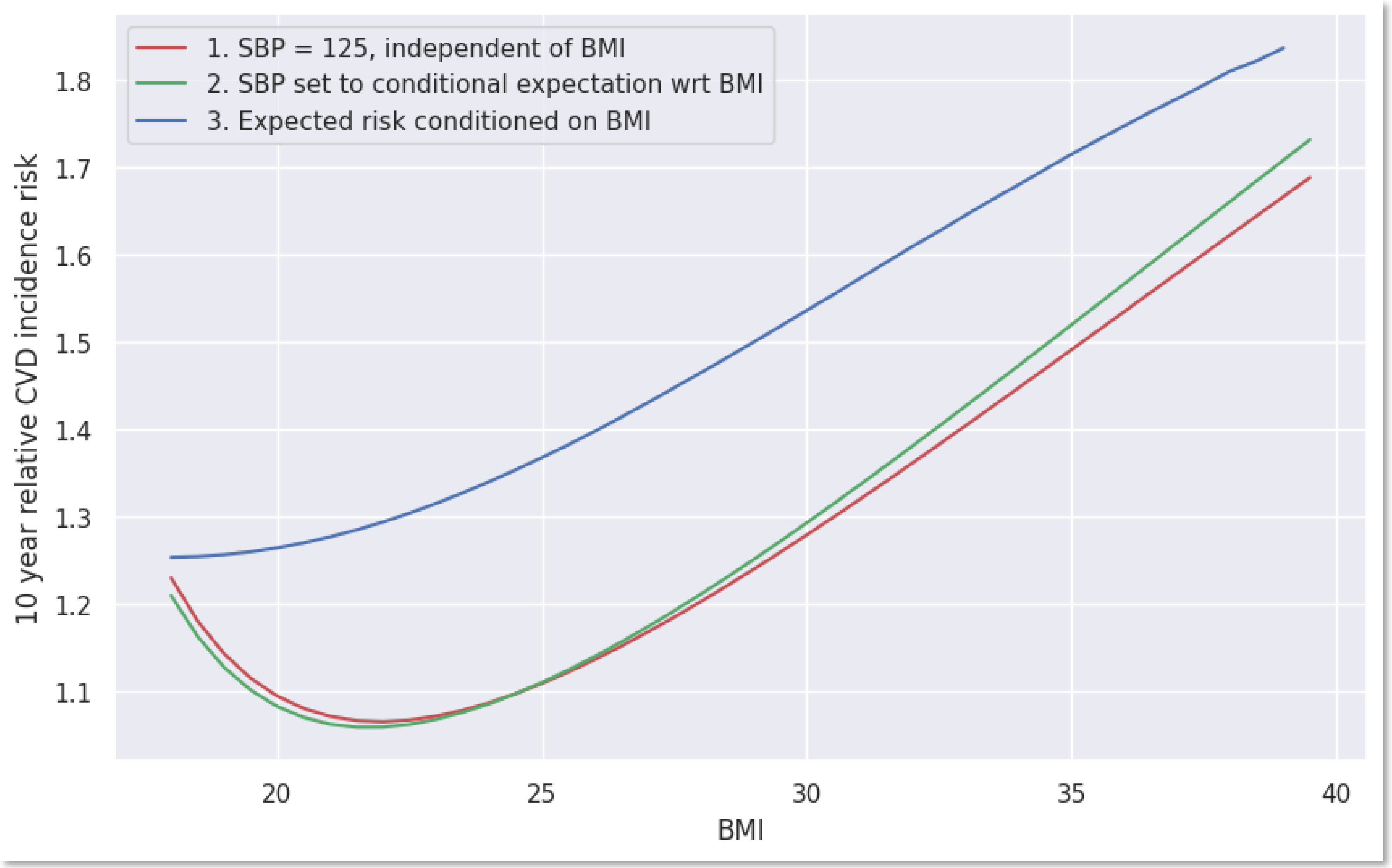


$$RR(\mathbf{x}) = \frac{\mu(\mathbf{x})}{\mu_{\mathrm{ref}}}$$

Country	Mean BMI females	Mean BMI males
Samoa	33.5	29.9
USA	28.8	28.8
UK	27.1	27.5
Germany	25.6	27.0
Italy	25.2	26.8
France	24.6	26.1
Switzerland	23.8	26.7
Japan	21.7	23.6

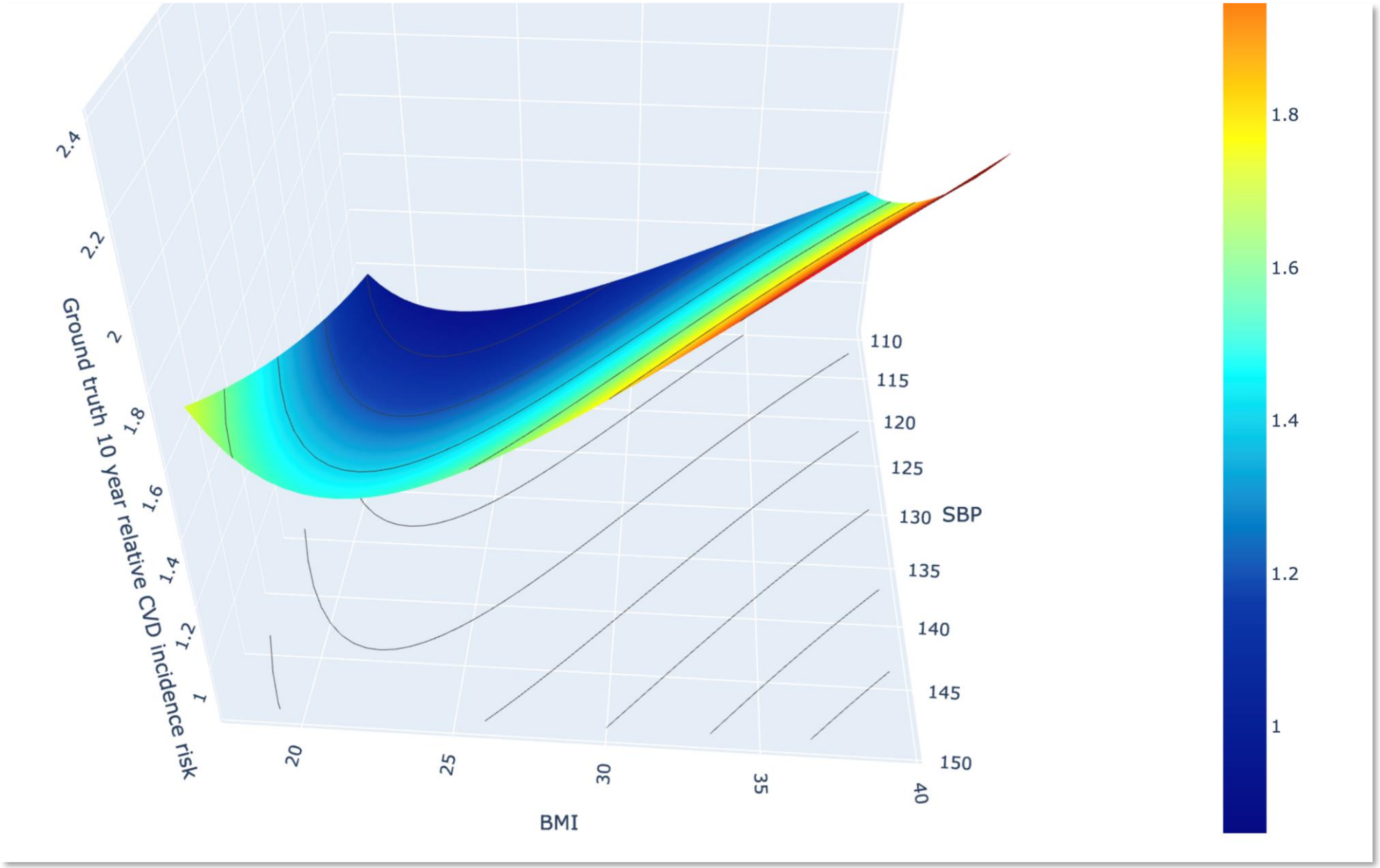
Country	Life exp. females	Life exp. females
Samoa	75.5	71.3
USA	81.5	76.5
UK	83.3	79.6
Germany	83.5	78.8
Italy	85.4	81.1
France	85.6	79.8
Switzerland	85.6	81.9
Japan	87.4	81.4

Various risk factors like BMI, systolic blood pressure (SBP) impact relative risk



1. What is the risk of a person with a given BMI and all other attributes equal to the reference person, $\mu(\mathrm{BMI}, \mathrm{SBP}_{\mathrm{ref}})$?
2. What is the risk of a person with a given BMI, and SBP set to the conditional expectation of SBP given BMI, $\mu(\mathrm{BMI}, \mathbb{E}[\mathrm{SBP}|\mathrm{BMI}])$?
3. What is the expected risk of a person with a given BMI, $\mathbb{E}[\mu(\mathbf{x})|\mathrm{BMI}]$?
4. What is the (causally implied) risk of the reference person when changing BMI, $\mathbb{E}[\mu(\mathbf{x})|do(\mathrm{BMI})]$?

Various risk factors like BMI, systolic blood pressure (SBP) impact relative risk



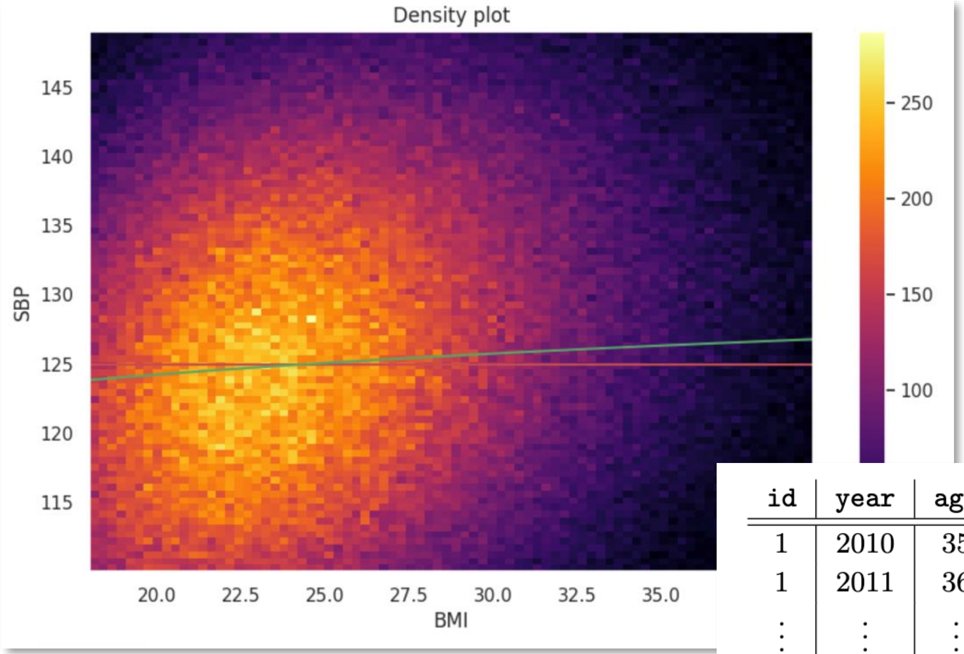
Creation of a synthetic health dataset

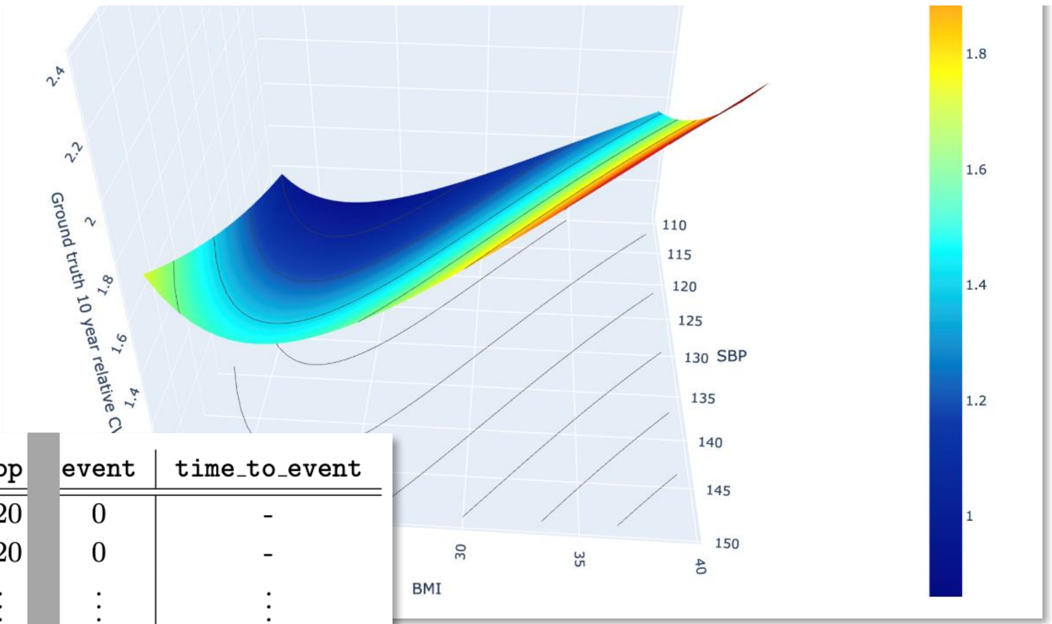
- **id**: an ID to uniquely identify a person,
- **year**: observation year of health information,
- **age**: age of the person at time **year**,
- **gender**: male (0)/female (1),
- **bmi**: body-mass-index (BMI), unit kg/m^2,
- **sbp**: systolic blood pressure (SBP), unit mmHg,
- **sd_sbp**: standard deviation of systolic blood pressure measurements, unit mmHg,
- **tcl_hdl_ratio**: total cholesterol level (TCL) divided by high-density lipoprotein level (HDL),
- **num1, num2, num3**: 3 generic numeric health risk factors without specifying their meaning explicitly, e.g., stepcounts, triglycerides, resting heartrate, etc.
- **binary**: a generic binary health risk factor, e.g., smokers, foreign born, etc.,

Creation of a synthetic health dataset

$$\begin{pmatrix} \texttt{sbp} \\ \log(\texttt{bmi}) \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 125 \\ 3.2 \end{pmatrix}, \begin{pmatrix} 15^2 & 15 \cdot 0.25\rho \\ 15 \cdot 0.25\rho & 0.25^2 \end{pmatrix} \right)$$

$$\begin{aligned} \mu^*(\boldsymbol{x}) &= \exp\Big(\log(\mathrm{QRISK3}(\boldsymbol{x})) + r(\texttt{age}) \\ &+ 16(\texttt{num1} - 0.5)^4 + 4(\texttt{num2} - 0.5)^2\texttt{num3} + \texttt{num3} + \texttt{binary} - 1.65 \Big) \end{aligned}$$



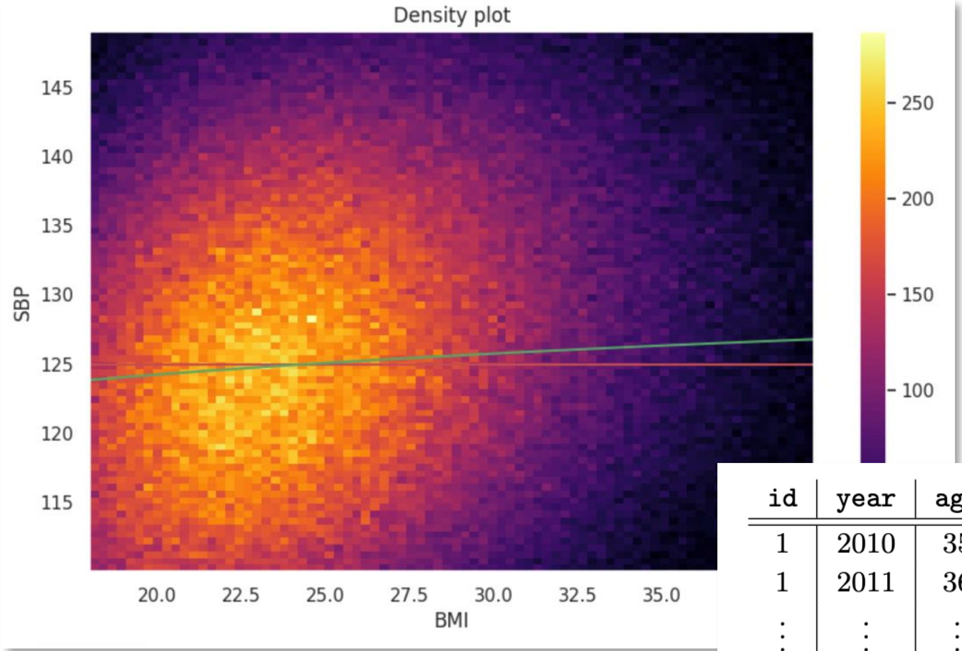


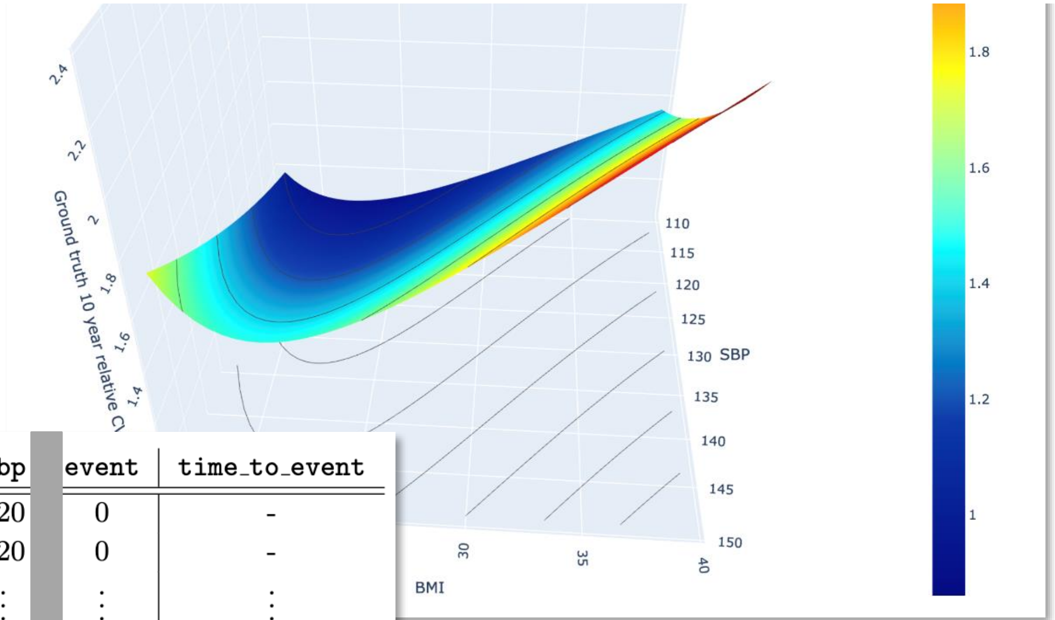
id	year	age	gender	bmi	sbp	…	event	time_to_event
1	2010	35	m	24	120		0	-
1	2011	36	m	24	120		0	-
⋮	⋮	⋮	⋮	⋮	⋮		⋮	⋮
1	2019	44	m	24	120		0	-
2	2010	35	m	33	145		0	7.5
2	2011	36	m	33	145		0	6.5
⋮	⋮	⋮	⋮	⋮	⋮		⋮	⋮
2	2017	44	m	33	145		1	0.5
3	2010	35	m	26	125		0	-
3	2011	36	m	26	125		0	-
⋮	⋮	⋮	⋮	⋮	⋮		⋮	⋮

Creation of a synthetic health dataset

$$\begin{pmatrix} \texttt{sbp} \\ \log(\texttt{bmi}) \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 125 \\ 3.2 \end{pmatrix}, \begin{pmatrix} 15^2 & 15\cdot 0.25\rho \\ 15\cdot 0.25\rho & 0.25^2 \end{pmatrix}\right)$$

$$\begin{aligned} \mu^*(\boldsymbol{x}) &= \exp\Big(\log(\mathrm{QRISK3}(\boldsymbol{x})) + r(\texttt{age}) \\ &+ 16(\texttt{num1}-0.5)^4 + 4(\texttt{num2}-0.5)^2\texttt{num3} + \texttt{num3} + \texttt{binary} - 1.65\Big) \end{aligned}$$





id	year	age	gender	bmi	sbp	...	event	time_to_event
1	2010	35	m	24	120		0	-
1	2011	36	m	24	120		0	-
⋮	⋮	⋮	⋮	⋮	⋮		⋮	⋮
1	2019	44	m	24	120		0	-
2	2010	35	m	33	145		0	7.5
2	2011	36	m	33	145		0	6.5
⋮	⋮	⋮	⋮	⋮	⋮		⋮	⋮
2	2017	44	m	33	145		1	0.5
3	2010	35	m	26	125		0	-
3	2011	36	m	26	125		0	-
⋮	⋮	⋮	⋮	⋮	⋮		⋮	⋮

Model 1: Logistic regression/generalized linear model (GLM)

$$\begin{aligned} \mu_1(\boldsymbol{x}) &= \frac{1}{1+\exp(-\beta_0-\beta_1 x_1-\cdots-\beta_k x_k)} \text{ , or equivalently} \\ \operatorname{logit}(\mu_1(\boldsymbol{x})) &= \beta_0+\beta_1 x_1+\cdots+\beta_k x_k \text{ ,} \end{aligned}$$

```
import statsmodels.formula.api as sm
log_reg = sm.logit(formula="E~SBP+BMI+I(BMI**2)", data=time_to_event_train).fit()
pred = log_reg.predict(time_to_event_test)
```

Odds/log-odds

$$\begin{aligned} \operatorname{odds}(y=1 \mid \boldsymbol{x}) &:= \frac{P(y=1 \mid \boldsymbol{x})}{P(y=0 \mid \boldsymbol{x})}=\frac{P(y=1 \mid \boldsymbol{x})}{1-P(y=1 \mid \boldsymbol{x})} \\ \log(\operatorname{odds}(y=1 \mid \boldsymbol{x})) &= \beta_0+\beta_1 x_1+\cdots+\beta_k x_k \text{ .} \end{aligned}$$

Odds ratios

$$\begin{aligned} \frac{\operatorname{odds}(y=1 \mid (x_1,\ldots,x_j+1,\ldots,x_k))}{\operatorname{odds}(y=1 \mid \boldsymbol{x})} &= \frac{\exp(\beta_0+\beta_1 x_1+\cdots+\beta_j(x_j+1)+\cdots+\beta_k x_k)}{\exp(\beta_0+\beta_1 x_1+\cdots+\beta_k x_k)} \\ &= \exp(\beta_j) \text{ .} \end{aligned}$$

id	year	age	gender	bmi	sbp	event	time_to_event
1	2010	35	m	24	120	0	-
1	2011	36	m	24	120	0	-
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	2019	44	m	24	120	0	-
2	2010	35	m	33	145	0	7.5
2	2011	36	m	33	145	0	6.5
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
2	2017	44	m	33	145	1	0.5
3	2010	35	m	26	125	0	-
3	2011	36	m	26	125	0	-
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Model 2: Cox regression

$$\boxed{h(t \mid \boldsymbol{x}) = h_0(t) \exp\left(\beta_1 x_1 + \cdots + \beta_k x_k\right)}$$

```
import lifelines as ll
cph = ll.CoxPHFitter()
cph.fit(time_to_event, "T", event_col="E", formula="SBP+BMI+I(BMI**2)")
pred = (1 - np.array(cph.predict_survival_function(time_to_event_test))[10,:])
```

From hazard rates to 10-year risk

$$\mu_2(\boldsymbol{x}) := 1 - \exp\left(- \int_0^{10} h(t \mid \boldsymbol{x})\ dt \right)$$

Hazard ratios

$$\frac{h_0(t) \exp\left(\beta_1 x_1 + \cdots + \beta_j (x_j + 1) + \cdots + \beta_k x_k\right)}{h_0(t) \exp\left(\beta_1 x_1 + \cdots + \beta_k x_k\right)} = \exp(\beta_j)$$

id	year	age	gender	bmi	sbp	event	time_to_event
1	2010	35	m	24	120	0	-
1	2011	36	m	24	120	0	-
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	2019	44	m	24	120	0	-
2	2010	35	m	33	145	0	7.5
2	2011	36	m	33	145	0	6.5
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
2	2017	44	m	33	145	1	0.5
3	2010	35	m	26	125	0	-
3	2011	36	m	26	125	0	-
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Model 3: Neural networks

```
import tensorflow as tf
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Input, Dense, Activation
model = Sequential([
        Input(shape=(10,)),
        Dense(256),
        Activation(tf.keras.activations.relu),
        Dense(128),
        Activation(tf.keras.activations.relu),
        Dense(64),
        Activation(tf.keras.activations.relu),
        Dense(1, activation = "sigmoid")
    ])
model.compile(optimizer = opt, loss = 'binary_crossentropy')
model.fit(X_train, y_train, batch_size = 64, epochs = 100, shuffle = True, validation_split = 0.20)
pred = model.predict(X_test).flatten()
```

$$\boxed{\begin{aligned}
\mu_3(\boldsymbol{x}) &= \boldsymbol{z}^{(4)} \circ \boldsymbol{z}^{(3)} \circ \boldsymbol{z}^{(2)} \circ \boldsymbol{z}^{(1)}(\boldsymbol{x}) \text{ , where} \\
z_j^{(\kappa)}(\boldsymbol{z}) &= \phi^{(\kappa)}\left(\beta_{0,j}^{(\kappa)} + \sum_{\ell=1}^{q_{\kappa-1}} \beta_{\ell,j}^{(\kappa)} z_\ell\right) \text{ , for } 1 \leq j \leq q_\kappa \text{ ,} \\
\phi^{(\kappa)}(x) &= \begin{cases} 1/(1+\exp(-x)) & , \kappa = 4, \\ x\mathbb{1}_{\{x\geq 0\}} & , \text{otherwise,} \end{cases} \\
(q_4, \ldots, q_0) &= (1, 64, 128, 256, 10) \text{ .}
\end{aligned}}$$

Layer	input size	output size	#parameters	input from layer κ
Fully-conn.	10	256	2′816	0
ReLU ϕ	256	256	0	1
Fully-conn.	256	128	32′896	1
ReLU ϕ	128	128	0	2
Fully-conn.	128	64	8′256	2
ReLU ϕ	64	64	0	3
Fully-conn.	64	1	65	3
Sigmoid output	1	1	0	4

id	year	age	gender	bmi	sbp		event	time_to_event
1	2010	35	m	24	120		0	-
1	2011	36	m	24	120		0	-
⋮	⋮	⋮	⋮	⋮	⋮		⋮	⋮
1	2019	44	m	24	120		0	-
2	2010	35	m	33	145		0	7.5
2	2011	36	m	33	145	...	0	6.5
⋮	⋮	⋮	⋮	⋮	⋮		⋮	⋮
2	2017	44	m	33	145		1	0.5
3	2010	35	m	26	125		0	-
3	2011	36	m	26	125		0	-
⋮	⋮	⋮	⋮	⋮	⋮		⋮	⋮

Model performance

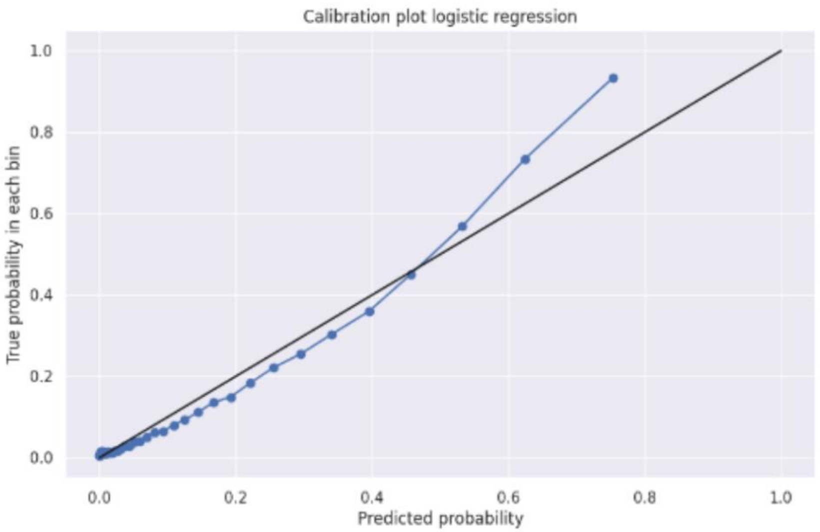
Table 3: Performance metrics on the test data subset of $\mathcal{D}_1$.

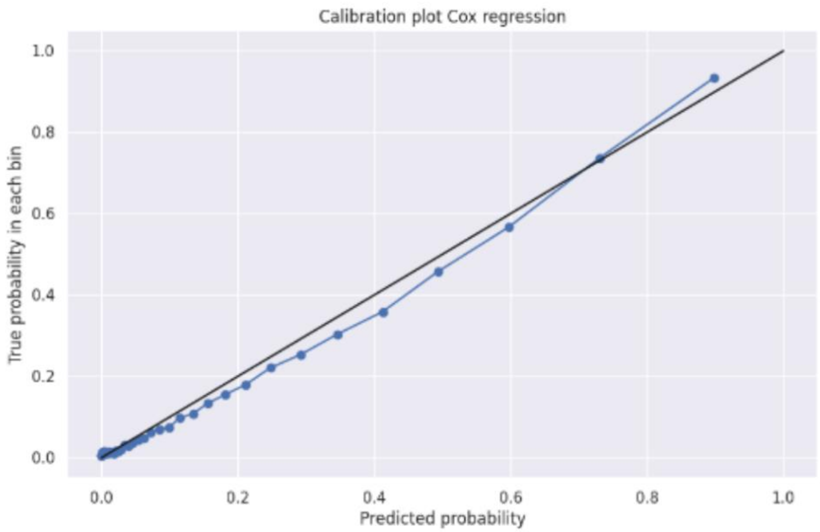
Performance metric	logistic regression $\mu_1(\boldsymbol{x})$	Cox regression $\mu_2(\boldsymbol{x})$	neural net $\mu_3(\boldsymbol{x})$
ROC AUC	56.17%	56.17%	56.04%
MSE wrt $\log(\mu^*(\boldsymbol{x}))$	0.0016	0.0016	0.0057
Logistic deviance	9223.88	9223.88	9227.72

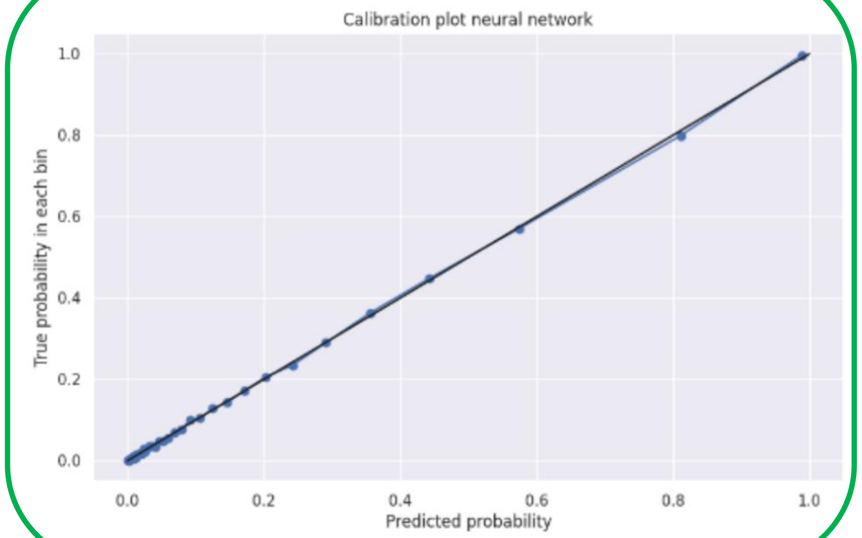
Table 4: Performance metrics on the test data subset of $\mathcal{D}_2$.

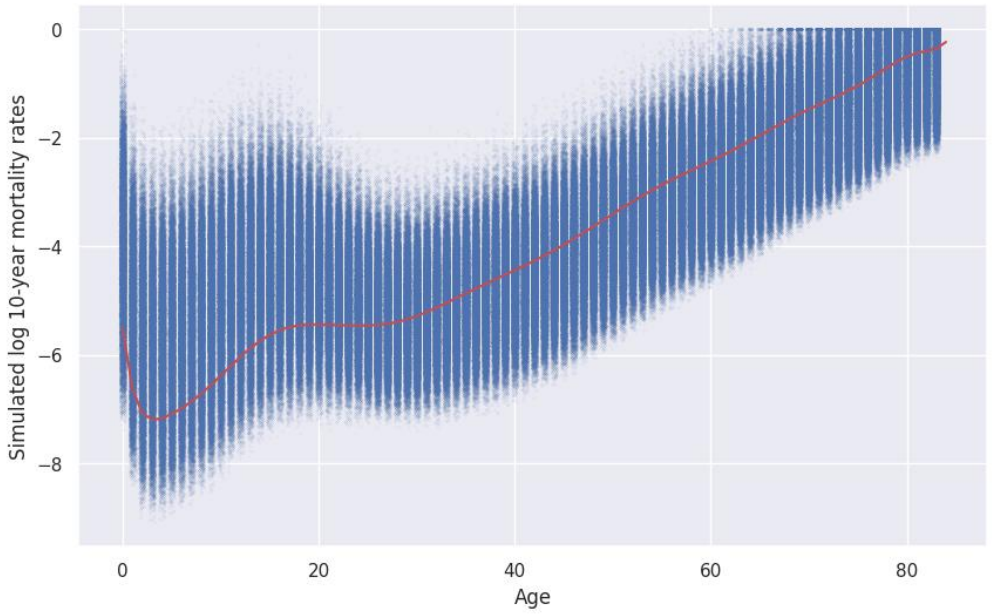
Performance metric	logistic regression $\mu_1(\boldsymbol{x})$	Cox regression $\mu_2(\boldsymbol{x})$	neural net $\mu_3(\boldsymbol{x})$
ROC AUC	90.54%	90.55%	92.05%
MSE wrt $\log(\mu^*(\boldsymbol{x}))$	1.74	1.75	0.11
Logistic deviance	85383	83994	75732

Model performance









Asymmetric cryptography (public/private key)

- Create a *shared secret s* for symmetric encryption (stream ciphers: Salsa20, RC4, ..., block ciphers: AES, DES, RC5, ...):

Alice: "Secret message" -> $m \in (\mathbb{Z}/2\mathbb{Z})^n$ -> $m + f(s)$ or $f(m, s)$ => **Bob**: $m + f(s) + f(s)$ or $f(f(m,s), s)$ -> "Secret message"

- Some examples:

1. RSA (Rivest, Shamir, Adleman, 1977): **Factoring** large integers *n* = *pq* (*n* public, *p*, *q* private)
2. ElGamal (1985): **Discrete logarithm**, (multiplicative) group G, usually $G \subset (\mathbb{Z}/p\mathbb{Z})^* =: \mathbb{F}_p^*$ of order *q* = (*p*-1)/2 with generator *g*, solve *x* = $\log_g h$ (*g*, G, *h* public, *x* private)
3. Elliptic curves methods (1985): **Discrete logarithm**, where group G is based on elliptic curves
4. Lattice based methods, e.g., LWE ("learning with errors", 2005): Solve $Ax + \varepsilon = b \bmod q$ for $x \in (\mathbb{Z}/q\mathbb{Z})^n$, where A is drawn uniformly from $(\mathbb{Z}/q\mathbb{Z})^{m\times n}$, $\varepsilon \in [-q/4, q/4]^m$ is drawn from a "non-trivial" distribution χ, and $b \in (\mathbb{Z}/q\mathbb{Z})^m$ (b, q, A public, x private)
5. Many more examples from NIST standardization proposals for post-quantum cryptography (factorization and discrete logarithm can be calculated very efficiently on quantum computers), e.g., CRYSTALS, 2018, while SIKE had to be removed from the list in August 2022.

RSA

- RSA, 1977 based on **Euler theorem**: $m^{\varphi(n)} = 1 \bmod n$ for $\gcd(m, n) = 1$, where $\varphi(n) = \#(\mathbb{Z}/n\mathbb{Z})^*$ (Euler's totient function)
  - $n = p = 7,\ \varphi(p) = p - 1$

0	1	2	3	4	5	6

  - $n = pq = 15,\ \varphi(pq) = (p-1)(q-1)$

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

  - Choose "random" p, q, d with $\gcd\big(d, \varphi(pq)\big) = 1$, calculate e with $ed = 1 \bmod \varphi(pq)$ with extended Euclidean algorithm, e, n public key, p, q, d private key
  - Encryption: message $m < n, m^e \bmod n$
  - Decryption: $m^{ed} = m \bmod n$
  - *Proof*: $m^{ed} = m^{k\varphi(n)+1} = m \bmod n$
  - Calculating d from e and n $\Leftrightarrow$ calculating $\varphi(n)$ $\Leftrightarrow$ **factoring** $n = pq$
  - *Proof idea*: "$\Leftarrow$" 1. $\varphi(pq) = (p-1)(q-1)$, 2. extended Euclidean algorithm $ed + b\varphi(pq) = \gcd\big(d, \varphi(pq)\big) = 1$
    "$\Rightarrow$" 1. $\varphi(pq) = -(p+q) + 1 \bmod n$, 2. $ed - 1 = k\varphi(n)$ sufficient to factor n (see, e.g., Miller, 1975, ERH)
  - There are attacks for, e.g., $q < p < 2q, 3d < n^{1/4}$ (Wiener, 1990) and several others (Zhang, 1999)
  - Homomorphic encryption RSA example: $(m_1 m_2)^e = m_1{}^e m_2{}^e \bmod n$, in general: enc(op$_1(m_1, m_2)$) = op$_2$(enc($m_1$), enc($m_2$))

Key take-aways

1. Limited data availability for sensitive personal (Life & Health) data in practice (e.g., nFADP, 1 September 2023)
2. For sufficiently large and dense datasets, ML/DL methods outperform traditional models, creating value for policyholders and insurance companies
3. Privacy preserving methods can help to access data