

CODECHECK certificate 2025-028

<https://doi.org/10.5281/zenodo.17533508>



Item	Value
Title of checked publication	Clinically Interpretable Survival Prediction in Primary Biliary Cholangitis with TreeSHAP and Gradient-Boosted Model
Author(s)	Emmanuel Pio Pastore
Reference	https://doi.org/10.71240/lcyc.66O7260
Codechecker(s)	Daniel Nüst
Date of check	2025-11-04 14:36:30
Summary	This code was straightforward to check and the reproduction is partially successful. Figure 1, 2, and 3 could be recreated without errors in the code but in part with differences beyond the variations due to re-computation. Table 2 could be reproduced within computational variation, but Table 3 could not be linked to any of the output files.
Repository	https://github.com/codecheckers/surv-tcav-pbc

Table 1: CODECHECK summary

Output	Comment	Size (b)
<code>results/split_metrics.csv</code>	Unclear which table/figure this corresponds to, possibly Table 3	3315
<code>results/summary_metrics.csv</code>	Table 2, with small numerical differences.	680
<code>figures/validation_metrics.png</code>	Figure 1	64952
<code>figures/tcav_bar.png</code>	Figure 3	79251
<code>figures/treeshap_summary.png</code>	Figure 2, with some differences in feature names and values.	292653

Table 2: Summary of output files generated

Summary

This code was straightforward to check. Figure 1, 2, and 3 could be recreated without errors in the code and in part with differences that are due to randomness in the workflow and not in contradicting the original results. Table 2 could be reproduced within computational variation, but Table 3 could not be linked to any of the output files. The reproduction is partially successful.

CODECHECKER notes

The GitHub repo <https://github.com/emmanuel6474/surv-tcav-pbc> is mentioned in the Code section of the article at <https://doi.org/10.71240/lcyc.66O7260>. The authors created an incorrectly formatted codecheck.yml file, but it provides a starting point to compile manifest information:

```
version: 1
requirements:
  - python<=3.12.8
  - -r requirements.txt
entrypoint:
  cmd: python
  args:
    - pbc_surv_tcav.py
    - --repeats
    - "5"
    - --no_shap
outputs:
  - results/split_metrics.csv
  - results/summary_metrics.csv
  - figures/validation_metrics.png
  - figures/tcav_bar.png
  - figures/calibration_tstar.png
optional:
  - figures/treeshap_summary.png
```

Based on these output files and other information, I updated the codecheck.yml file in the check fork at <https://github.com/codecheckers/surv-tcav-pbc>. This check is based on the commit c7433e6afdfcb7f28029fec94f566bbc75637286.

The actual workflow execution was straightforward based on the instructions in the README file, which is complemented by the information from the Materials section of the paper on used hardware:

“The computational environment is fully specified. Analyses were conducted in Python 3.11 on a workstation with an AMD Ryzen 7 5700X CPU. Key libraries included: xgboost, pandas, numpy, scikit-learn, lifelines, matplotlib, and shap.”

I created a local environment with the suggested commands, albeit based on my local Python version 3.10.12. The installation of the pinned dependencies worked without issues.

Code Is written in Python in a single script file `pbc_surv_tcav.py`. The code is split up into many functions, but uses a lot of abbreviated variable names, which makes it hard to follow the logic of the code as an external reviewer. There are no comments in the code, and no docstrings for the functions. Crucially, the README lacks a clear description of which tables and figures in the paper correspond to which output files from the code execution.

Then, I first run the “Faster run” mode as suggested in the README file but including the optional treeshap figure, with the following output (manually line-broken for readability):

```
.venv) daniel@laptop-nuest ~/git/codecheck/lifecycle-journal-codechecks/48/surv-tcav-pbc [main]$ python \
  pbc_surv_tcav.py --repeats 5
{"dataset": "PBC-276", "source": "https://vincentarelbundock.github.io/Rdatasets/csv/survival/pbc.csv",
 "sha256": "797ea9b6abfec34297ef07f361a2e0bddd90c3c2def180adf8547ea30e75b613",
 "n_rows": 276,
 "features": 17,
 "rows_dropped_missing": 142,
 "censoring_rate": 0.598}
--- Reference SOTA (same dataset/task; user-supplied) ---
No --sota_json provided. Add one to print verified reference numbers for EXACTLY the same protocol.
[run 01 seed=42] XGB-AFT C=0.887 | IBS=0.363 | tau=4427.000
[run 02 seed=43] XGB-AFT C=0.813 | IBS=0.340 | tau=4256.000
[run 03 seed=44] XGB-AFT C=0.840 | IBS=0.351 | tau=4427.000
[run 04 seed=45] XGB-AFT C=0.835 | IBS=0.373 | tau=4556.000
[run 05 seed=46] XGB-AFT C=0.850 | IBS=0.332 | tau=4256.000
[info] per-split metrics saved -> results/split_metrics.csv
[info] summary metrics saved -> results/summary_metrics.csv
--- FAIR COMPARISON SUMMARY (PBC-276, 25x 80/20; IBS[0, tau_eff], KM on TRAIN, G>gmin) ===
model C_mean C_sd C_boot_lo C_boot_hi IBS_mean IBS_sd IBS_boot_lo IBS_boot_hi gap_C_to_best gap_IBS_to_best
  xgb  0.845  0.027    0.825    0.869   0.352  0.017      0.339    0.365      0.000     0.202
  cox  0.837  0.027    0.814    0.858   0.150  0.041      0.124    0.186      0.008     0.000
  waft 0.824  0.029    0.802    0.846   0.158  0.040      0.130    0.192      0.021     0.008
[info] validation plot saved -> figures/validation_metrics.png
```

```

[info] TreeSHAP saved -> figures/treeshap_summary.png
== Surv-TCAV (last split) - directional effect on ==
[TCAV] cholestasis      Δ = -342.362840 (-1283.930714,+604.287570) Δ /SD( ) = -0.104 (-0.390,+0.184) pos_rate=0.12
[TCAV] coagulopathy     Δ = +22.269302 (-252.168345,+318.759151) Δ /SD( ) = +0.007 (-0.077,+0.097) pos_rate=0.27
[TCAV] low_albumin      Δ = -154.954063 (-515.330334,+179.620646) Δ /SD( ) = -0.047 (-0.157,+0.055) pos_rate=0.25
[TCAV] older_age        Δ = -412.217567 (-693.317137,-176.420951) Δ /SD( ) = -0.125 (-0.211,-0.054) pos_rate=0.25
[TCAV] clinical_complications Δ = -325.000553 (-1240.637570,+618.042643) Δ /SD( ) = -0.099 (-0.377,+0.188) pos_rate=0.35
[info] TCAV bar saved -> figures/tcav_bar.png
[info] calibration plot saved -> figures/calibration_tstar.png
== Reporting notes (TRIPOD-lite) ==
* Internal validation (25x 80/20)
* Event of interest: death (status==2). Transplant treated as censored.

```

This completes in a few minutes without errors, and most expected output files are created. Only `figures/calibration_tstar.png` is missing, even though the log output indicates it should have been created. After consultation with the author (see below), the figure is not used in the submission and is not generated due to a small sample size.

Now I am manually comparing the created output files to the figures and tables in the paper:

- `figures/tcav_bar.png` appears to correspond to Figure 3 in the paper with matching plot style, title, and axis labels, though considerable different data, likely because of the limited repeats;
- `figures/treeshap_summary.png` appears to correspond to Figure 2 in the paper with matching plot style, title, and axis labels, and the data seems to have roughly similar patterns, though the features are sorted differently likely because of different “mean absolute SHAP value” because of the limited repeats;
- `figures/validation_metrics.png` appears to correspond to Figure 1 in the paper with matching plot style, title, and axis labels, though considerable fewer data points and different patterns, likely because of the limited repeats;
- `results/split_metrics.csv` likely corresponds to one of the tables in the paper, but it is not possible to determine from the face of the values (due to different number of repeats) or the very short variable names which table this is;
- `results/summary_metrics.csv` likely corresponds to one of the tables in the paper, but it is not possible to determine from the face of the values (due to different number of repeats) or the very short variable names which table this is;

To possibly fix the discrepancies, I re-ran the code with the default configuration, and this also completed within a few minutes (manually line-broken for readability):

```

(.venv) daniel@laptop-nuest ~/git/codecheck/lifecycle-journal-codechecks/48/surv-tcav-pbc [main]$ python pbc_surv_tcav.py
{"dataset": "PBC-276", "source": "https://vincentarelbundock.github.io/Rdatasets/csv/survival/pbc.csv",
 "sha256": "797ea9b6abfc34297ef07f361a2e0bfdd90c3c2def180adf8547ea30e75b613",
 "n_rows": 276,
 "features": 17,
 "rows_dropped_missing": 142,
 "censoring_rate": 0.598}
--- Reference SOTA (same dataset/task; user-supplied) ---
No --sota_json provided. Add one to print verified reference numbers for EXACTLY the same protocol.
[run 01 seed=42] XGB-AFT C=0.887 | IBS=0.363 | tau=4427.000
[run 02 seed=43] XGB-AFT C=0.813 | IBS=0.340 | tau=4256.000
[run 03 seed=44] XGB-AFT C=0.840 | IBS=0.351 | tau=4427.000
[run 04 seed=45] XGB-AFT C=0.835 | IBS=0.373 | tau=4556.000
[run 05 seed=46] XGB-AFT C=0.850 | IBS=0.332 | tau=4256.000
[run 06 seed=47] XGB-AFT C=0.840 | IBS=0.350 | tau=4509.000
[run 07 seed=48] XGB-AFT C=0.745 | IBS=0.384 | tau=4500.000
[run 08 seed=49] XGB-AFT C=0.817 | IBS=0.342 | tau=4256.000
[run 09 seed=50] XGB-AFT C=0.817 | IBS=0.319 | tau=3933.000
[run 10 seed=51] XGB-AFT C=0.908 | IBS=0.346 | tau=4256.000
[run 11 seed=52] XGB-AFT C=0.906 | IBS=0.353 | tau=4256.000
[run 12 seed=53] XGB-AFT C=0.813 | IBS=0.368 | tau=4556.000
[run 13 seed=54] XGB-AFT C=0.879 | IBS=0.374 | tau=4523.000
[run 14 seed=55] XGB-AFT C=0.816 | IBS=0.321 | tau=4191.000
[run 15 seed=56] XGB-AFT C=0.848 | IBS=0.387 | tau=4509.000
[run 16 seed=57] XGB-AFT C=0.844 | IBS=0.363 | tau=4556.000
[run 17 seed=58] XGB-AFT C=0.864 | IBS=0.366 | tau=4556.000
[run 18 seed=59] XGB-AFT C=0.793 | IBS=0.365 | tau=4427.000
[run 19 seed=60] XGB-AFT C=0.889 | IBS=0.363 | tau=4523.000
[run 20 seed=61] XGB-AFT C=0.813 | IBS=0.355 | tau=4365.000
[run 21 seed=62] XGB-AFT C=0.861 | IBS=0.361 | tau=4556.000
[run 22 seed=63] XGB-AFT C=0.805 | IBS=0.383 | tau=4500.000

```

```

[run 23 seed=64] XGB-AFT C=0.798 | IBS=0.354 | tau=4256.000
[run 24 seed=65] XGB-AFT C=0.824 | IBS=0.342 | tau=4365.000
[run 25 seed=66] XGB-AFT C=0.807 | IBS=0.361 | tau=4556.000
[info] per-split metrics saved -> results/split_metrics.csv
[info] summary metrics saved -> results/summary_metrics.csv
== FAIR COMPARISON SUMMARY (PBC-276, 25x 80/20; IBS[0, tau_eff], KM on TRAIN, G>gmin) ===
model C_mean C_sd C_boot_lo C_boot_hi IBS_mean IBS_sd IBS_boot_lo IBS_boot_hi gap_C_to_best gap_IBS_to_best
xgb 0.837 0.038 0.822 0.851 0.357 0.018 0.349 0.363 0.000 0.207
cox 0.829 0.034 0.815 0.842 0.149 0.039 0.136 0.164 0.007 0.000
waft 0.818 0.037 0.803 0.832 0.155 0.036 0.143 0.169 0.019 0.006
[info] validation plot saved -> figures/validation_metrics.png
[info] TreeSHAP saved -> figures/treeshap_summary.png
== Surv-TCAV (last split) - directional effect on ===
[TCAV] cholestasis Δ = -569.489481 (-1644.770115,+433.051415) Δ /SD( ) = -0.179 (-0.516,+0.136) pos_rate=0.12
[TCAV] coagulopathy Δ = -309.125045 (-529.181518,-100.936478) Δ /SD( ) = -0.097 (-0.166,-0.032) pos_rate=0.26
[TCAV] low_albumin Δ = -583.421913 (-932.153546,-238.641145) Δ /SD( ) = -0.183 (-0.293,-0.075) pos_rate=0.25
[TCAV] older_age Δ = -536.163012 (-969.125221,-139.228449) Δ /SD( ) = -0.168 (-0.304,-0.044) pos_rate=0.25
[TCAV] clinical_complications Δ = -713.050014 (-1648.918126,+228.438344) Δ /SD( ) = -0.224 (-0.518,+0.072) pos_rate=0.34
[info] TCAV bar saved -> figures/tcav_bar.png
[info] calibration plot saved -> figures/calibration_tstar.png
== Reporting notes (TRIPOD-lite) ===
* Internal validation (25x 80/20)
* Event of interest: death (status==2). Transplant treated as censored.

```

summary_metrics.csv now has values that fit Table 2 relatively closely, so that is successfully reproduced within computational variation. I still cannot link split_metrics.csv to Table 3 based on the values .treeshap_summary.png now looks much closer to the original Figure 2, though there are still some curious difference, such as “stage_2.0” bein missing in my reproduction, but the reproduction having a “spiders_1.0” that is missing from the original.

To summarize, while the code runs without errors and produces most of the expected output files, but they cannot be perfectly matched to the original results. Some of these differences may go beyond to be expected numerical differences from my perspective.

After consultation with the author (see comment on GitHub), the author claims the deviations are small and expected due to stochastic nature of the workflow.

The differences you noted are all in the same direction as in the paper, vary only slightly in magnitude, and the 95% confidence intervals still overlap completely. These are well within the range expected for stochastic models and have no impact on the interpretation or conclusions. In the SHAP summary plots, the feature names do not actually change in a meaningful way. The small differences visible near the bottom of the figure involve features with very low impact, close to statistical noise, so it is natural that they may move slightly in or out of the importance ranking. The overall clusters of points are visually and numerically very similar, and the main features at the top of the ranking vary only minimally because they are much more significant.

I have no reason to question this statement, yet would invite topical reviewers to assess the interpretation of the differences independently.

Recommendations

I suggest to the authors to consider the following suggestions for their next publication or workflow:

- Clearly name output files in a way that makes it easy to link them to figures and tables in the paper (e.g. “figure_2_treeshap_summary.png” instead of “treeshap_summary.png”);
- Try to set seeds or other random number generation initializations in a way that allows exact reproduction of results;

Manifest files

split_metrics.csv

Summary statistics of tabular data:

```
-- Data Summary -----
      Values
Name          read.csv(path)
Number of rows 25
Number of columns 11

-----  
Column type frequency:
 logical           2
 numeric          9

-----  
Group variables     None

-- Variable type: logical -----
skim_variable n_missing complete_rate mean count
1 c_rsf            25             0  NaN ":" "
2 ibs_rsf          25             0  NaN ":" "

-- Variable type: numeric -----
skim_variable n_missing complete_rate    mean      sd
1 run              0               1   13     7.36
2 seed             0               1   54     7.36
3 tau              0               1 4403.   160.
4 c_xgb            0               1   0.837  0.0384
5 ibs_xgb          0               1   0.357  0.0178
6 c_cox            0               1   0.829  0.0343
7 ibs_cox          0               1   0.149  0.0385
8 c_waft           0               1   0.818  0.0368
9 ibs_waft         0               1   0.155  0.0356

      p0      p25      p50      p75      p100 hist
1   1       7      13      19      25
2   42      48      54      60      66
3 3933    4256    4427    4523    4556
4   0.745  0.813  0.835  0.861  0.908
5   0.319  0.346  0.361  0.366  0.387
6   0.759  0.808  0.839  0.851  0.884
7   0.102  0.124  0.136  0.160  0.240
8   0.744  0.794  0.808  0.849  0.879
9   0.103  0.132  0.147  0.165  0.233
```

summary_metrics.csv

Summary statistics of tabular data:

```
-- Data Summary -----
      Values
Name          read.csv(path)
Number of rows    3
Number of columns 11

-----  
Column type frequency:
 character        1
 numeric         10

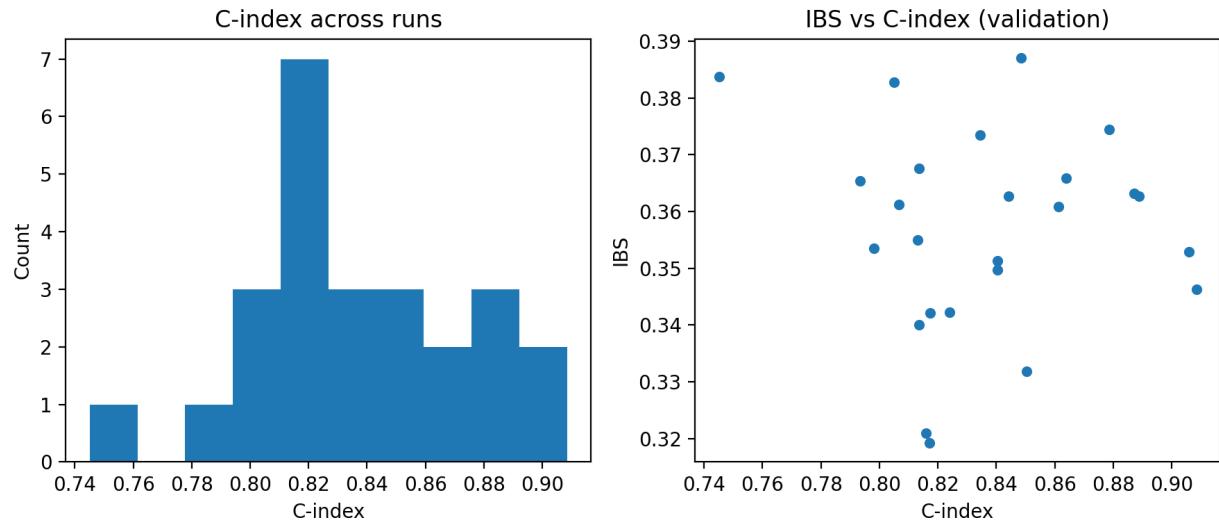
-----  
Group variables     None

-- Variable type: character -----
skim_variable n_missing complete_rate min max empty
1 model            0             1   3   4   0
n_unique whitespace
1           3           0

-- Variable type: numeric -----
skim_variable  n_missing complete_rate   mean      sd
1 C_mean          0             1  0.828  0.00944
2 C_sd            0             1  0.0365 0.00206
3 C_boot_lo       0             1  0.814  0.00961
4 C_boot_hi       0             1  0.842  0.00960
5 IBS_mean         0             1  0.220  0.118
6 IBS_sd           0             1  0.0306 0.0112
7 IBS_boot_lo     0             1  0.209  0.121
8 IBS_boot_hi     0             1  0.232  0.114
9 gap_C_to_best    0             1  0.00866 0.00944
10 gap_IBS_to_best 0             1  0.0710 0.118
      p0    p25    p50    p75    p100 hist
1 0.818 0.824 0.829 0.833 0.837
2 0.0343 0.0355 0.0368 0.0376 0.0384
3 0.803 0.809 0.815 0.819 0.822
4 0.832 0.837 0.842 0.847 0.851
5 0.149 0.152 0.155 0.256 0.357
6 0.0178 0.0267 0.0356 0.0371 0.0385
7 0.136 0.139 0.143 0.246 0.349
8 0.164 0.166 0.169 0.266 0.363
9 0 0.00363 0.00725 0.0130 0.0187
10 0 0.00285 0.00571 0.106 0.207
```

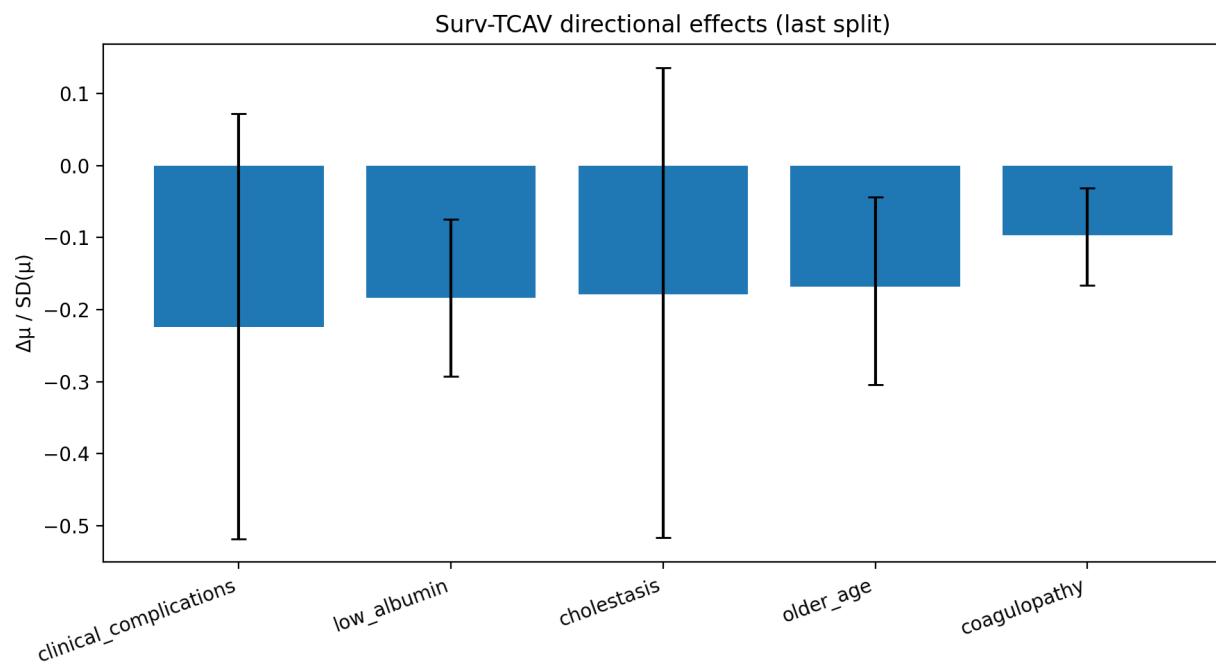
validation_metrics.png

Comment: Figure 1



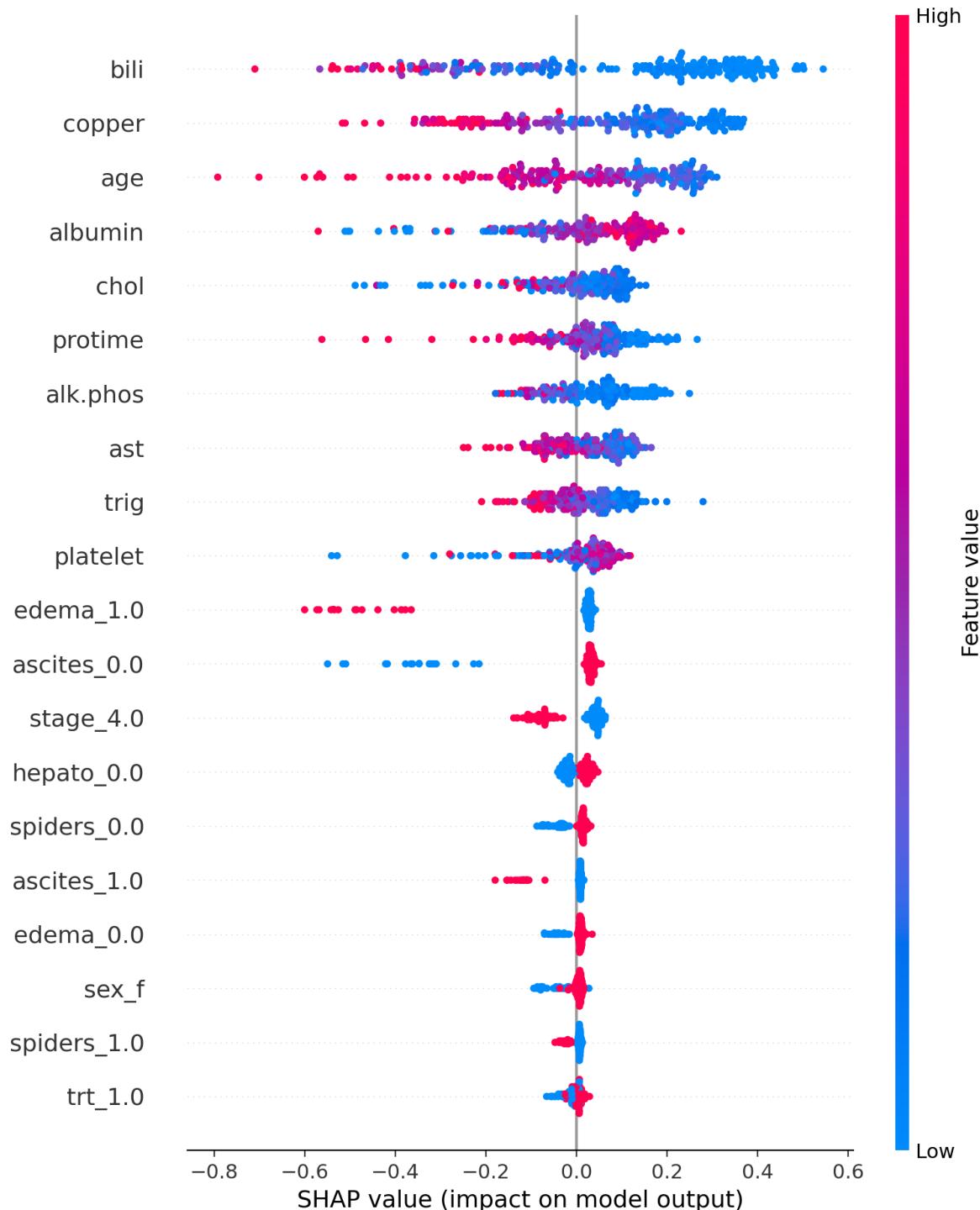
tcav_bar.png

Comment: Figure 3



treeshap_summary.png

Comment: Figure 2, with some differences in feature names and values.



Citing this document

Daniel Nüst (2025). CODECHECK Certificate 2025-028. Zenodo. <https://doi.org/10.5281/zenodo.17533508>

About CODECHECK

This certificate confirms that the codechecker could independently reproduce the results of a computational analysis given the data and code from a third party. A CODECHECK does not check whether the original computation analysis is correct. However, as all materials required for the reproduction are freely available by following the links in this document, the reader can then study for themselves the code and data.

About this document

This document was created using R Markdown using the `codecheck` R package. `make codecheck.pdf` will regenerate the report file.

```
sessionInfo()

## R version 4.5.1 (2025-06-13)
## Platform: x86_64-pc-linux-gnu
## Running under: Ubuntu 22.04.5 LTS
##
## Matrix products: default
## BLAS:    /usr/lib/x86_64-linux-gnublas/libblas.so.3.10.0
## LAPACK: /usr/lib/x86_64-linux-gnulapack/liblapack.so.3.10.0 LAPACK version 3.10.0
##
## locale:
## [1] LC_CTYPE=en_US.UTF-8      LC_NUMERIC=C
## [3] LC_TIME=de_DE.UTF-8      LC_COLLATE=en_US.UTF-8
## [5] LC_MONETARY=de_DE.UTF-8   LC_MESSAGES=en_US.UTF-8
## [7] LC_PAPER=de_DE.UTF-8     LC_NAME=C
## [9] LC_ADDRESS=C              LC_TELEPHONE=C
## [11] LC_MEASUREMENT=de_DE.UTF-8 LC_IDENTIFICATION=C
##
## time zone: Europe/Berlin
## tzcode source: system (glibc)
##
## attached base packages:
## [1] stats      graphics   grDevices datasets  utils
## [6] methods    base
##
## other attached packages:
## [1] readr_2.1.5      tibble_3.3.0      xtable_1.8-4
## [4] yaml_2.3.10      rprojroot_2.1.1   knitr_1.50
## [7] codecheck_0.22.0  parsedate_1.3.2   R.cache_0.17.0
## [10] gh_1.5.0
##
## loaded via a namespace (and not attached):
## [1] xfun_0.53        rdflib_0.2.9      bspm_0.5.7
## [4] tzdb_0.5.0       vctrs_0.6.5       tools_4.5.1
## [7] generics_0.1.4    parallel_4.5.1   curl_7.0.0
## [10] pkgconfig_2.0.3   pdftools_3.6.0    R.oo_1.27.1
## [13] skimr_2.2.1      redland_1.0.17-18 lifecycle_1.0.4
## [16] git2r_0.36.2     compiler_4.5.1   atom4R_0.3-4
```

```
## [19] stringr_1.5.2      repr_1.1.7        keyring_1.4.1
## [22] htmltools_0.5.8.1   crayon_1.5.3      pillar_1.11.1
## [25] whisker_0.4.1       tidyrr_1.3.1       R.utils_2.13.0
## [28] cachem_1.1.0        zen4R_0.10.2     tidyselect_1.2.1
## [31] zip_2.3.3           digest_0.6.37    stringi_1.8.7
## [34] dplyr_1.1.4          purrrr_1.1.0     fastmap_1.2.0
## [37] cli_3.6.5            magrittr_2.0.4    base64enc_0.1-3
## [40] triebeard_0.4.1     utf8_1.2.6       XML_3.99-0.19
## [43] crul_1.6.0           withr_3.0.2       osfr_0.2.9
## [46] bit64_4.6.0-1        roxygen2_7.3.3    rmarkdown_2.29
## [49] httr_1.4.7            bit_4.6.0         qpdf_1.4.1
## [52] askpass_1.2.1         R.methodsS3_1.8.2 hms_1.1.3
## [55] memoise_2.0.1         evaluate_1.0.5    urltools_1.7.3.1
## [58] rlang_1.1.6            Rcpp_1.1.0        glue_1.8.0
## [61] httpcode_0.3.0         formatR_1.14     xml2_1.4.0
## [64] fauxpas_0.5.2         rorcid_0.7.0      rstudioapi_0.17.1
## [67] vroom_1.6.6            jsonlite_2.0.0    plyr_1.8.9
## [70] R6_2.6.1              fs_1.6.6
```