Machine Learning Applied to Hepatocellular Carcinoma

Sangil Lee, Yujing Lu, Josh Tomiyama

2022-04-28

Section 1

Background HCC

HepatoCellular Carcinoma (HCC)

- HepatoCellular Carcinoma (HCC) 6th most frequently diagnosed cancer.

HepatoCellular Carcinoma (HCC)

- HepatoCellular Carcinoma (HCC) 6th most frequently diagnosed cancer.
- Data mining approach to tailor evaluation and treatment for HCC are limited in the literature.

HepatoCellular Carcinoma (HCC)

- HepatoCellular Carcinoma (HCC) 6th most frequently diagnosed cancer.
- Data mining approach to tailor evaluation and treatment for HCC are limited in the literature.
- Using the HCC dataset, we undertook the data mining approach to evaluate the patient level factors to identify those who are at risk of one year mortality.

Section 2

Data Summary

Data Summary

- Clinical data of 165 pts with HCC (demographic, risk factors, lab data, and survival features)
- 49 features from HCC clinical practice guidelines (Table 1)
- About 80% male, 74% had alcohol related liver disease, 27% had hepatitis B, 21% had hepatitis C, and 90% had cirrhosis.
- Missing data represents 10.22% of the whole dataset and only eight patients have complete information in all fields (4.85%).
- Missing data were imputed using various methods, then step_nzv was used to remove the variables that have near zero variability.
- The target variable is the survival at 1 year, coded as 0 (dies) and 1 (lives).

Data Summary Table

Table 1: HCC Data Summary of Selected Variables

	Overall (N=165)
gender	
female	32 (19.4%)
male	133 (80.6%)
symptom	
no	53 (36.1%)
yes	94 (63.9%)
N-Miss	18
alc	
no	43 (26.1%)
yes	122 (73.9%)

Section 3

Random Forest Model

Description

- RF is a modification of bagging that is comparable to boosting but is simpler to train and tune- works by building and averaging a large collection of de-correlated trees.

Description

- RF is a modification of bagging that is comparable to boosting but is simpler to train and tune- works by building and averaging a large collection of de-correlated trees.
- 10 fold CV

Description

- RF is a modification of bagging that is comparable to boosting but is simpler to train and tune- works by building and averaging a large collection of de-correlated trees.
- 10 fold CV
- Tuned Hyperparameter Parameters with Bayesian Optimization

Description

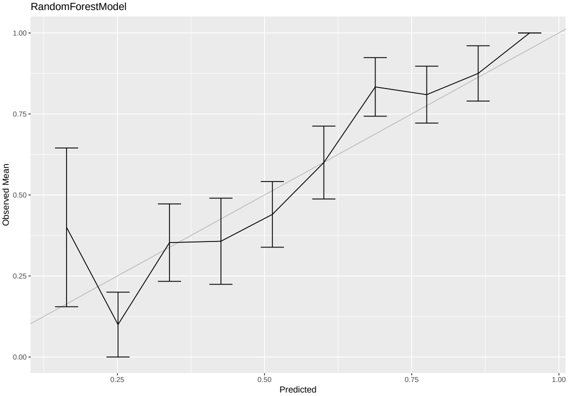
- RF is a modification of bagging that is comparable to boosting but is simpler to train and tune- works by building and averaging a large collection of de-correlated trees.
- 10 fold CV
- Tuned Hyperparameter Parameters with Bayesian Optimization
- Applied and tuned KNN imputation and Correlation Filter

Estimated Performance

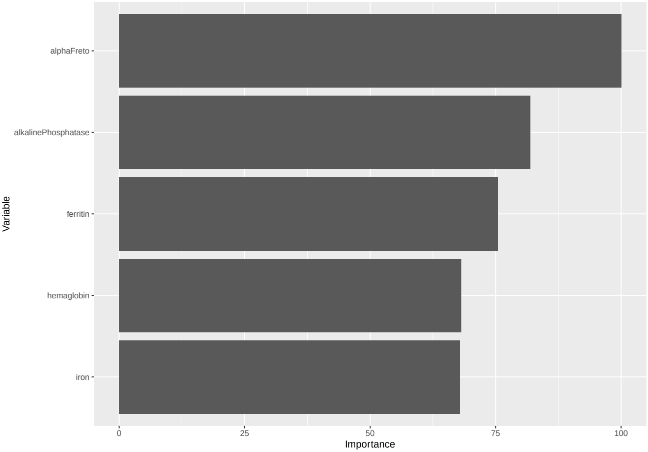
Table 2: RF results with knn imputation and corr filter

Metric	Mean	Median	SD	Min	Max	NA
Brier	0.175	0.175	0.046	0.115	0.245	0
Accuracy	0.720	0.719	0.103	0.562	0.875	0
Kappa	0.390	0.383	0.220	0.143	0.714	0
ROC AUC	0.799	0.808	0.137	0.608	0.983	0
Sensitivity	0.824	0.800	0.146	0.500	1.000	0
Specificity	0.555	0.619	0.184	0.333	0.833	0

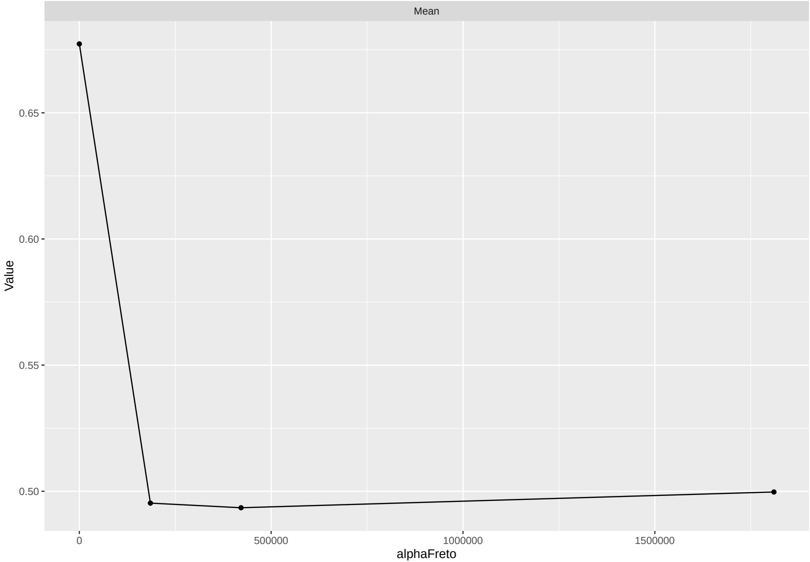
Calibration Plots



Variable Importance



Partial Dependence



Section 4

XGBoost Model

Description

Input: training set $\{(x_i, y_i)\}_{i=1}^{N}$, a differentiable loss function $L(y, F(x))$, a number of weak learners M and a learning rate α.

1. Initialize model with a constant value:
   $\hat{f}_{(0)}(x) = argmin_{\theta} \sum_{i=1}^{N} L(y_i, \theta)$.
2. For $m = 1, ..., M$:
   a Compute gradients and hessians: $\hat{g}_m(x_i) = \left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f(x)=\hat{f}_{m-1}(x)}$
   $\hat{h}_m(x_i) = \left[\frac{\partial^2 L(y_i, f(x_i))}{\partial f(x_i)^2}\right]_{f(x)=\hat{f}_{m-1}(x)}$.

   b Fit a base learner using the training set $\left\{x_i, -\frac{\hat{g}_m(x_i)}{\hat{h}_m(x_i)}\right\}_{i=1}^{N}$ by solving the optimization problem below:
   $\hat{\phi}_m = argmin_{\phi \in \Phi} \sum_{i=1}^{N} \frac{1}{2} \hat{h}_m(x_i) \left[-\frac{\hat{g}_m(x_i)}{\hat{h}_m(x_i)} - \phi(x_i)\right]^2$.

   c Update the model:
   $\hat{f}_m(x) = \hat{f}_{(m-1)}(x) + \alpha \hat{\phi}_m$.
3. Output $\hat{f}(x) = \hat{f}_{(M)}(x)$.

Description

- XGBoost includes different regularization penalties to prevent overfitting. It can be run on multiple CPU cores and servers to save training time.

Description

- XGBoost includes different regularization penalties to prevent overfitting. It can be run on multiple CPU cores and servers to save training time.
- Tuning parameters: number of boosting iterations M, shrinkage of variable weights at each iteration to prevent overfitting η, maximum tree depth.

Description

- XGBoost includes different regularization penalties to prevent overfitting. It can be run on multiple CPU cores and servers to save training time.
- Tuning parameters: number of boosting iterations M, shrinkage of variable weights at each iteration to prevent overfitting η, maximum tree depth.
- Tuning inputs: KNN with the number of neighbors, correlation among variables, principal component analysis proportion of variance to be retained.

Estimated Performance

Table 3: XGBoost results with knn imputation and feature selection using correlation

Metric	Mean	Median	SD	Min	Max	NA
Brier	0.221	0.219	0.086	0.081	0.344	0
Accuracy	0.701	0.688	0.096	0.562	0.889	0
Kappa	0.360	0.353	0.210	0.097	0.766	0
ROC AUC	0.768	0.775	0.116	0.584	0.961	0
Sensitivity	0.774	0.800	0.116	0.600	0.909	0
Specificity	0.586	0.619	0.187	0.333	0.857	0

- Tried six models with different combinations of imputation methods and feature selection methods. All models are not doing well for specificity.

Estimated Performance

Table 4: XGBoost results with knn imputation and feature selection using correlation

Metric	Mean	Median	SD	Min	Max	NA
Brier	0.221	0.219	0.086	0.081	0.344	0
Accuracy	0.701	0.688	0.096	0.562	0.889	0
Kappa	0.360	0.353	0.210	0.097	0.766	0
ROC AUC	0.768	0.775	0.116	0.584	0.961	0
Sensitivity	0.774	0.800	0.116	0.600	0.909	0
Specificity	0.586	0.619	0.187	0.333	0.857	0

- KNN_corr appears to do better among models for its highest accuracy and kappa, second highest sensitivity and specificity, reasonable brier and roc auc.

Estimated Performance

Table 5: XGBoost results with knn imputation and feature selection using correlation

Metric	Mean	Median	SD	Min	Max	NA
Brier	0.221	0.219	0.086	0.081	0.344	0
Accuracy	0.701	0.688	0.096	0.562	0.889	0
Kappa	0.360	0.353	0.210	0.097	0.766	0
ROC AUC	0.768	0.775	0.116	0.584	0.961	0
Sensitivity	0.774	0.800	0.116	0.600	0.909	0
Specificity	0.586	0.619	0.187	0.333	0.857	0

- P-values are not significant.

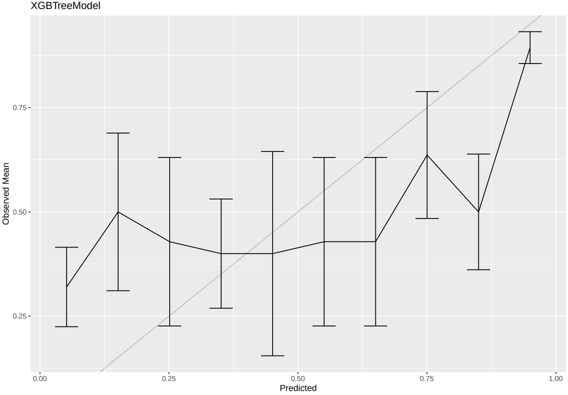
Estimated Performance

Table 5: XGBoost results with knn imputation and feature selection using correlation

Metric	Mean	Median	SD	Min	Max	NA
Brier	0.221	0.219	0.086	0.081	0.344	0
Accuracy	0.701	0.688	0.096	0.562	0.889	0
Kappa	0.360	0.353	0.210	0.097	0.766	0
ROC AUC	0.768	0.775	0.116	0.584	0.961	0
Sensitivity	0.774	0.800	0.116	0.600	0.909	0
Specificity	0.586	0.619	0.187	0.333	0.857	0

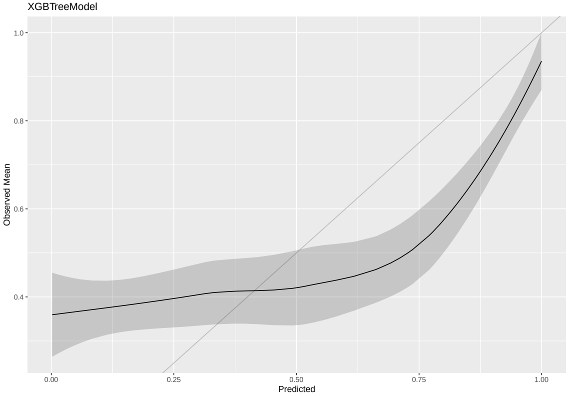
- P-values are not significant.
- Bayesian optimization selected $\eta = 0.1$, maximum tree depth $= 7$, number of boosting iterations $= 106$, number of features $= 44$.

Calibration Plot



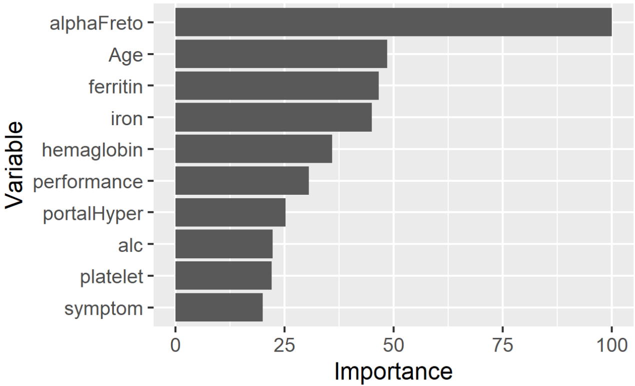
- Some fluctuations.

Smoothed Calibration Plot

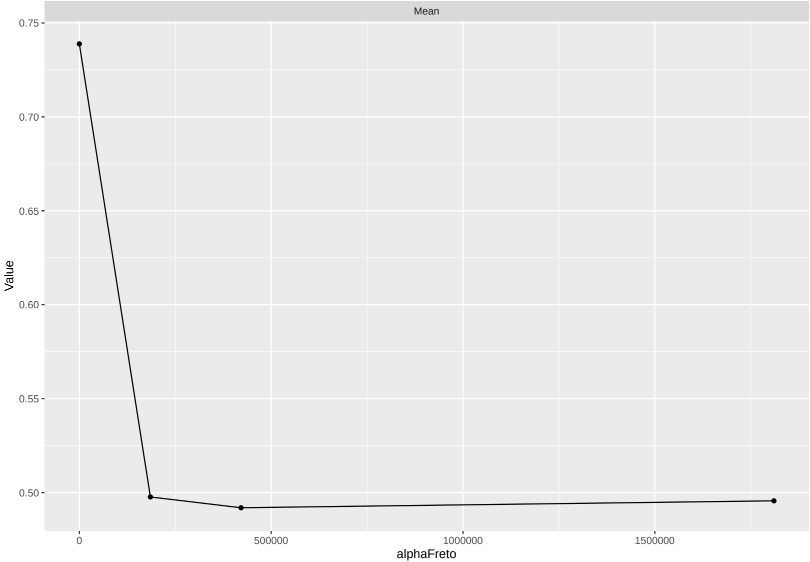


- Smoothed calibration curve shows false negatives on the lower left part and some false positives on the upper right part. This is consistent to the low specificity and relatively high sensitivity of the model.

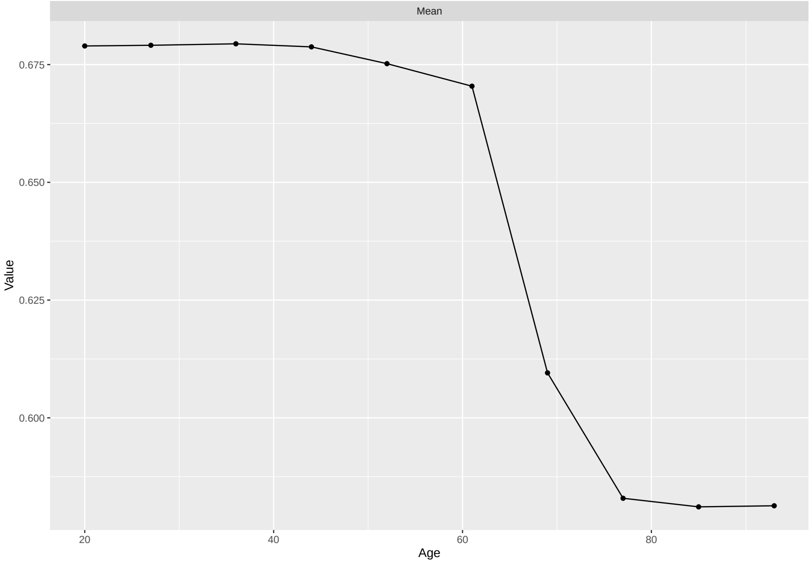
Variable Importance



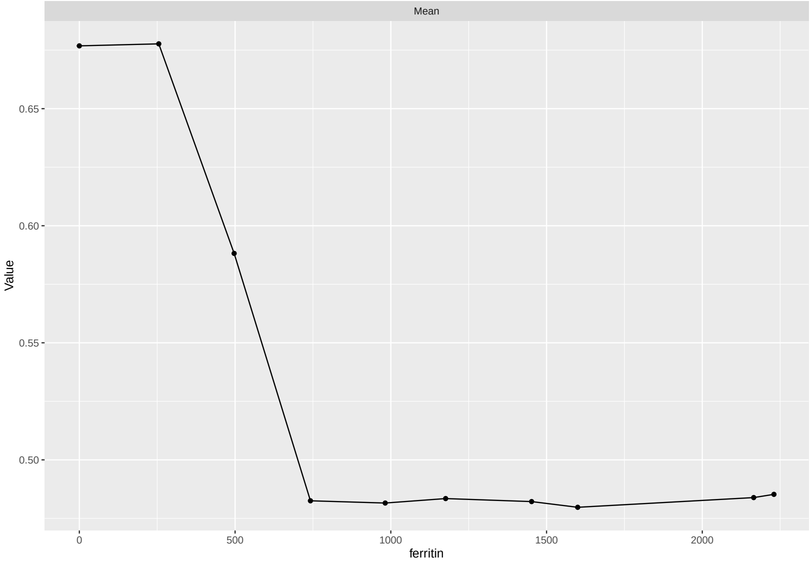
Partial Dependence: alphaFreto



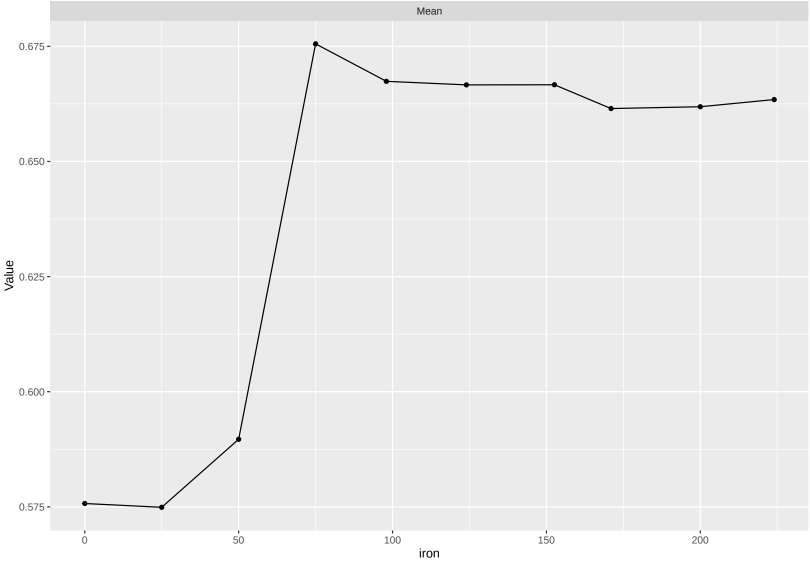
Partial Dependence: Age



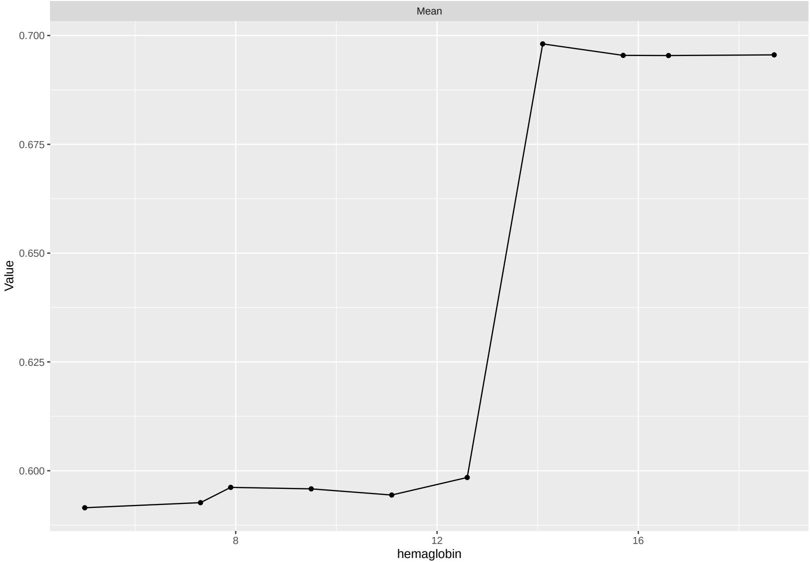
Partial Dependence: ferritin



Partial Dependence: iron



Partial Dependence: hemaglobin



Section 5

Support Vector Machine Model

Description

- A support vector machine attempts to draw a non-linear boundary to classify the outcome by transforming the predictors using basis functions

Description

- A support vector machine attempts to draw a non-linear boundary to classify the outcome by transforming the predictors using basis functions
- In this transformed space, find linear boundaries using support vector classifier.

Description

- A support vector machine attempts to draw a non-linear boundary to classify the outcome by transforming the predictors using basis functions
- In this transformed space, find linear boundaries using support vector classifier.
- We just need to know the kernel function on the basis functions, don't need to know basis function itself.

Description

- A support vector machine attempts to draw a non-linear boundary to classify the outcome by transforming the predictors using basis functions
- In this transformed space, find linear boundaries using support vector classifier.
- We just need to know the kernel function on the basis functions, don't need to know basis function itself.
- Radial Basis kernel function: $\exp(-\sigma^2\|x - x'\|^2)$; $\sigma = 0.0123873$

Description

- A support vector machine attempts to draw a non-linear boundary to classify the outcome by transforming the predictors using basis functions
- In this transformed space, find linear boundaries using support vector classifier.
- We just need to know the kernel function on the basis functions, don't need to know basis function itself.
- Radial Basis kernel function: $\exp(-\sigma^2\|x - x'\|^2)$; $\sigma =$ 0.0123873
- Cost of misclassification C =0.8956493

Description

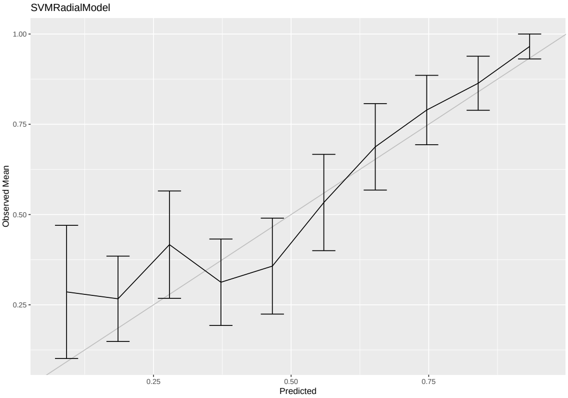
- A support vector machine attempts to draw a non-linear boundary to classify the outcome by transforming the predictors using basis functions
- In this transformed space, find linear boundaries using support vector classifier.
- We just need to know the kernel function on the basis functions, don't need to know basis function itself.
- Radial Basis kernel function: $\exp(-\sigma^2\|x - x'\|^2)$; $\sigma = 0.0123873$
- Cost of misclassification C $=$0.8956493
- imputation using KNN with neighbors $= 4$

Estimated Performance

Table 6: SVMRad results with knn imputation

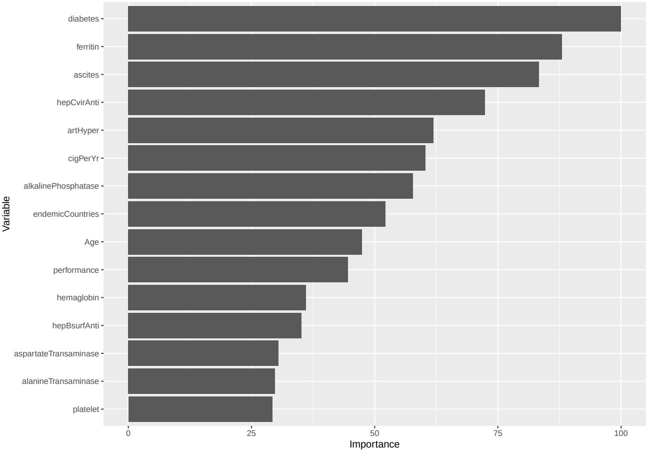
Metric	Mean	Median	SD	Min	Max	NA
Brier	0.176	0.166	0.057	0.101	0.287	0
Accuracy	0.756	0.764	0.119	0.562	0.938	0
Kappa	0.482	0.503	0.254	0.097	0.871	0
ROC AUC	0.818	0.829	0.101	0.600	0.967	0
Sensitivity	0.803	0.809	0.107	0.600	0.909	0
Specificity	0.681	0.643	0.190	0.500	1.000	0

Calibration Plots



- Decently calibrated. Low probabilities have many false negatives.

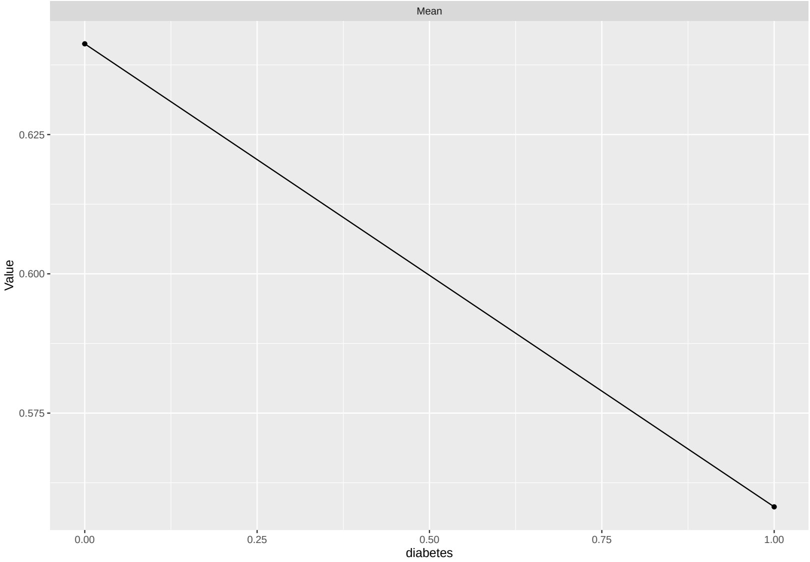
Variable Importance



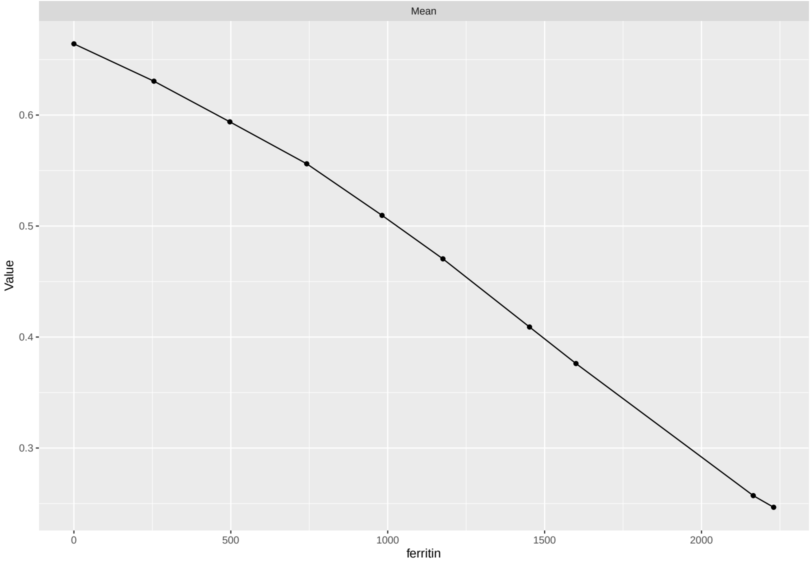
Variable Importance

- Symptoms (performance)
- Indicators of liver injury/disease/infection (hepBsurfAnti, hepCvirAnti,aspartateTransaminase, alkalinePhosphatase, ferritin, totalProteins, ascites, hemoglobin)
- Biological Characteristics (age, portalVeinThromb)
- Risk factors (diabetes, artHyper)
- behavioral/demographic (endemicCountries, cigPerYr)

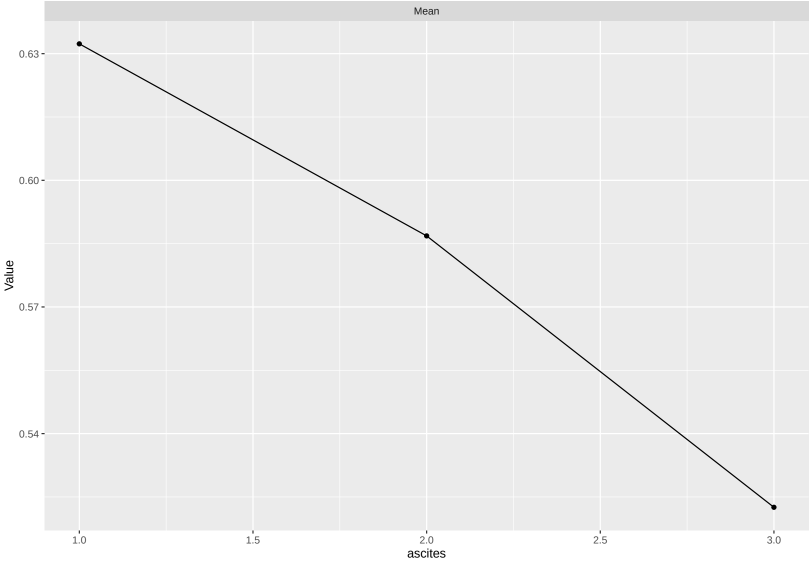
Partial Dependence: diabetes



Partial Dependence: ferritin



Partial Dependence ascites



Section 6

Final Model choice

Criteria

- Final model was chosen by the model with the highest Sensitivity

Table 7: Comparison of Sensitivity

	Metric	Mean	Median	SD	Min	Max
RandomForest	Sensitivity	0.824	0.800	0.146	0.5	1.000
XGBTree	Sensitivity	0.774	0.800	0.116	0.6	0.909
SVMRadial	Sensitivity	0.803	0.809	0.107	0.6	0.909

Criteria

- Final model was chosen by the model with the highest Sensitivity
- This corresponds to correctly identifying surviving patients (wrt cutoff = 0.5)

Table 7: Comparison of Sensitivity

	Metric	Mean	Median	SD	Min	Max
RandomForest	Sensitivity	0.824	0.800	0.146	0.5	1.000
XGBTree	Sensitivity	0.774	0.800	0.116	0.6	0.909
SVMRadial	Sensitivity	0.803	0.809	0.107	0.6	0.909

Criteria

- Final model was chosen by the model with the highest Sensitivity
- This corresponds to correctly identifying surviving patients (wrt cutoff = 0.5)
- Minimizes incorrectly telling a patient they will die

Table 7: Comparison of Sensitivity

	Metric	Mean	Median	SD	Min	Max
RandomForest	Sensitivity	0.824	0.800	0.146	0.5	1.000
XGBTree	Sensitivity	0.774	0.800	0.116	0.6	0.909
SVMRadial	Sensitivity	0.803	0.809	0.107	0.6	0.909

Section 7

References

References I

Auguie, Baptiste. 2017. *gridExtra: Miscellaneous Functions for "Grid" Graphics*. https://CRAN.R-project.org/package=gridExtra.

Bache, Stefan Milton, and Hadley Wickham. 2020. *Magrittr: A Forward-Pipe Operator for r*. https://CRAN.R-project.org/package=magrittr.

Chen, Tianqi, and Carlos Guestrin. 2016. "XGBoost: A Scalable Tree Boosting System." In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–94. KDD '16. New York, NY, USA: ACM. https://doi.org/10.1145/2939672.2939785.

Corporation, Microsoft, and Stephen Weston. 2022a. *doSNOW: Foreach Parallel Adaptor for the 'Snow' Package*. https://CRAN.R-project.org/package=doSNOW.

References II

Corporation, Microsoft, and Steve Weston. 2022b. *doParallel: Foreach Parallel Adaptor for the 'Parallel' Package*. https://CRAN.R-project.org/package=doParallel.

Heinzen, Ethan, Jason Sinnwell, Elizabeth Atkinson, Tina Gunderson, and Gregory Dougherty. 2021. *Arsenal: An Arsenal of 'r' Functions for Large-Scale Statistical Summaries*. https://CRAN.R-project.org/package=arsenal.

Kuhn, Max, and Hadley Wickham. 2022. *Recipes: Preprocessing and Feature Engineering Steps for Modeling*. https://CRAN.R-project.org/package=recipes.

R Core Team. 2021. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

References III

Santos, Miriam Seoane, Pedro Henriques Abreu, Pedro J. García-Laencina, Adélia Simão, and Armando Carvalho. 2015. "A New Cluster-Based Oversampling Method for Improving Survival Prediction of Hepatocellular Carcinoma Patients." *Journal of Biomedical Informatics* 58: 49–59. https://doi.org/https://doi.org/10.1016/j.jbi.2015.09.012.

Smith, Brian J. 2021. *MachineShop: Machine Learning Models and Tools*. https://cran.r-project.org/package=MachineShop.

Therneau, Terry M. 2021. *A Package for Survival Analysis in r*. https://CRAN.R-project.org/package=survival.

Wickham, Hadley, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, et al. 2019. "Welcome to the tidyverse." *Journal of Open Source Software* 4 (43): 1686. https://doi.org/10.21105/joss.01686.

References IV

Wickham, Hadley, and Maximilian Girlich. 2022. *Tidyr: Tidy Messy Data*. https://CRAN.R-project.org/package=tidyr.

Xie, Yihui. 2014. "Knitr: A Comprehensive Tool for Reproducible Research in R." In *Implementing Reproducible Computational Research*, edited by Victoria Stodden, Friedrich Leisch, and Roger D. Peng. Chapman; Hall/CRC. http://www.crcpress.com/product/isbn/9781466561595.

———. 2015. *Dynamic Documents with R and Knitr*. 2nd ed. Boca Raton, Florida: Chapman; Hall/CRC. https://yihui.org/knitr/.

———. 2021. *Knitr: A General-Purpose Package for Dynamic Report Generation in r*. https://yihui.org/knitr/.