

Machine Learning for time-to-event data:

Methodological advances and practical limitations

Quantitative Staff Network (QSN) Talk

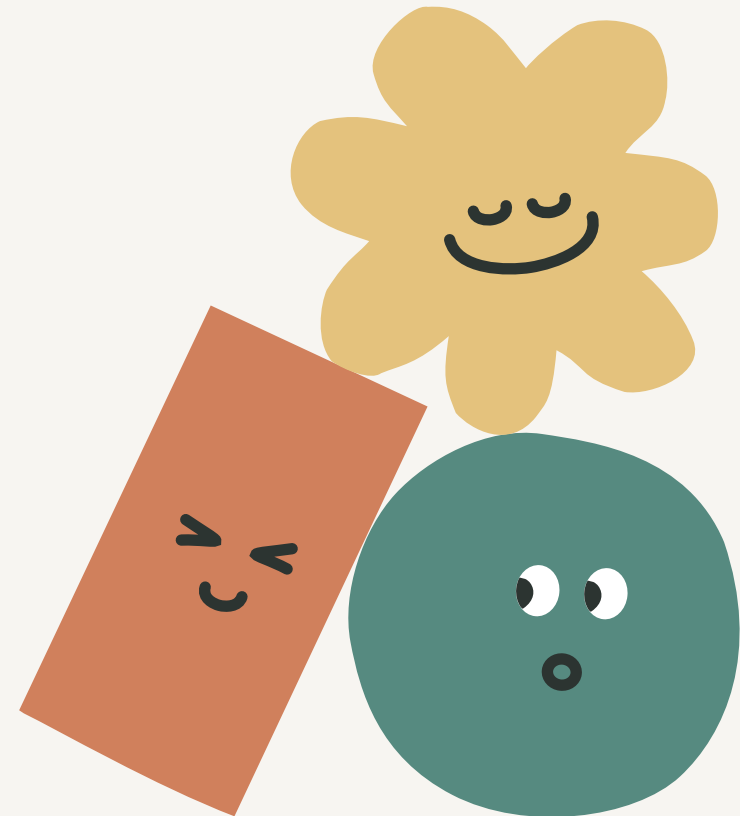
About Me

- I am a biostatistician with a background in experimental physics.
- My current position supports AI data services – we use AI tools to extract and analyze data from typical EHR information to image analysis and information extraction out of unstructured notes.
- I am skeptical of AI methods and the way that AI tools are portrayed as a replacement for people.



“The pessimist complains about the wind; the optimist expects it to change; the realist adjusts the sails”

William Arthur Ward



Outline

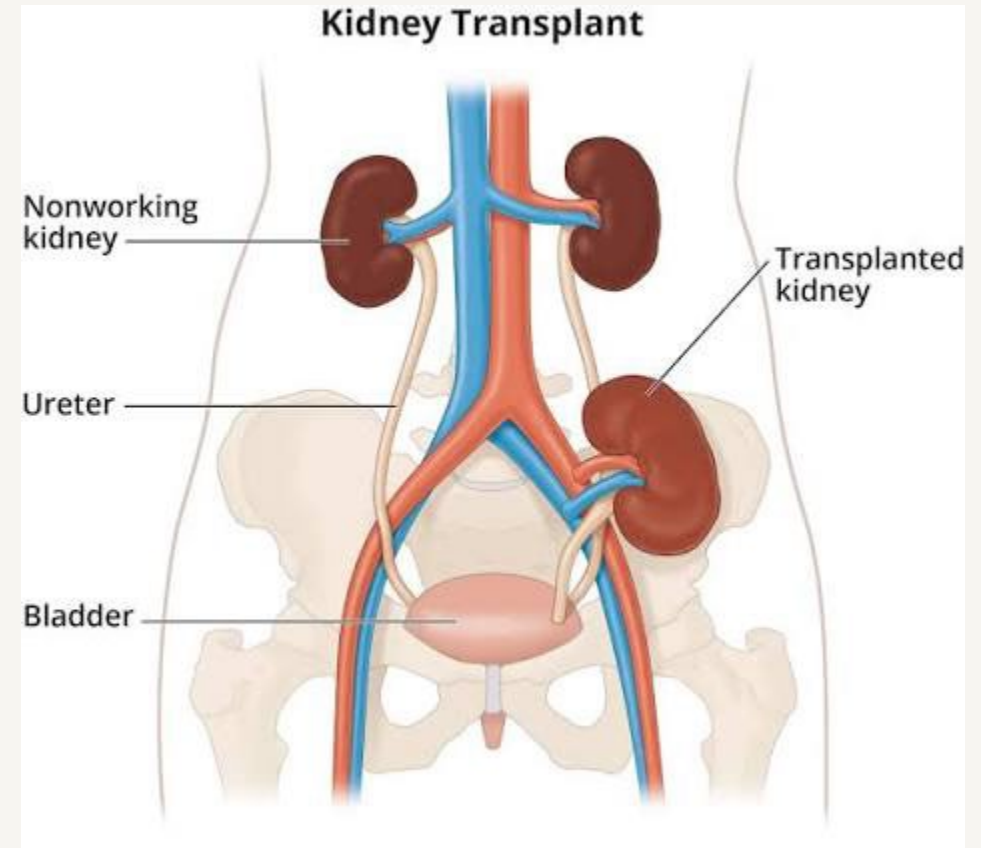
- A brief introduction to time-to-event analyses with examples
- Standard methods
 - Assumptions of linearity and model specification and proportionality
- Machine learning
 - Survival trees, random forests, and other methods
- How to evaluate and compare models
- What is the cost of using machine learning methods?
- A proposal for using machine learning.
- Conclusions
- Questions

Time-to-event analysis

- A time-to-event analysis looks at a combination of presence/absence of some event as well as the time to either that event or a last contact.
- Examples:
 - For students entering a particular educational program, what interventions or strategies improve retention and reduce drop-out?
 - For two types of fans, if we run 100 of each for 36 hours straight, how can we summarize failure rate?
 - In kidney transplant, how does age of the recipient impact the time to kidney failure post-transplant?

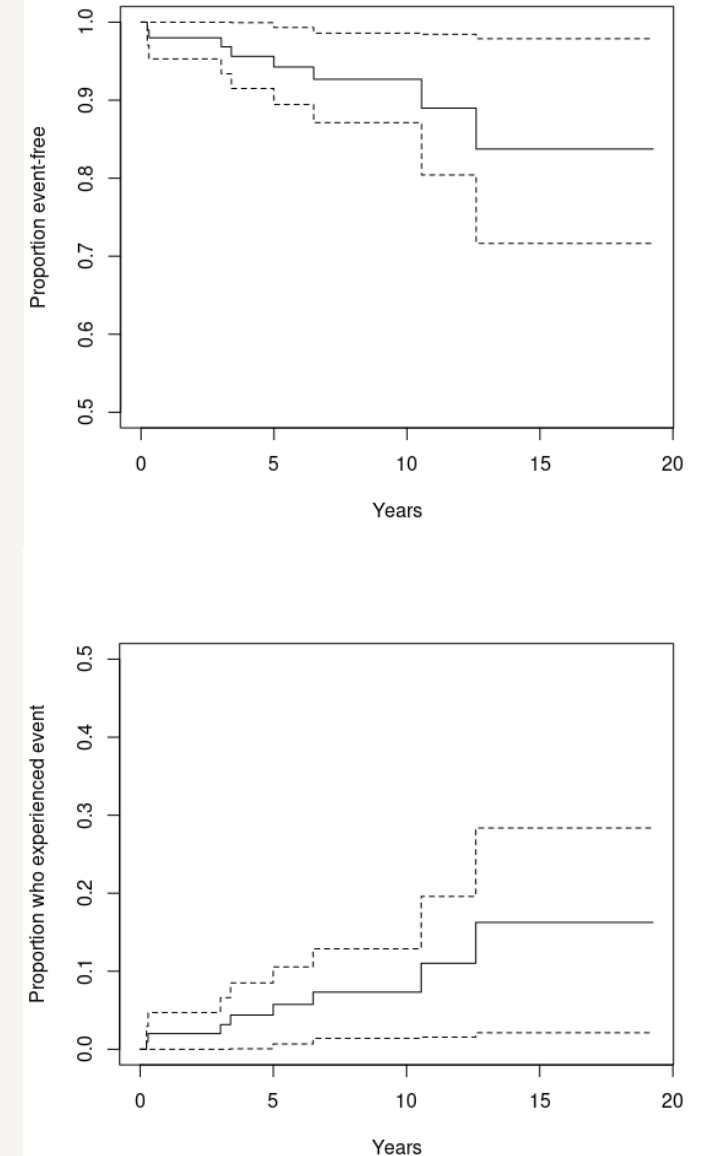
Kidney Transplant

- Kidney transplantation is a solution for patients whose kidneys have failed.
- The most common causes of kidney failure in the US are diabetes and hypertension, but there are many more causes.
- There are about 100,000 patients on the kidney transplant waitlist and between 25,000 and 30,000 transplants per year in the US.
- The major endpoint of interest to kidney transplant is graft failure – loss of the transplanted kidney. This is about 50% at 15 years post-transplant.



Major concepts in time-to-event analysis

- Think about trying to keep track of a proportion over time. What is the proportion of your sample that has experienced the event of interest.
- The **number at risk** is basically the denominator in an estimate of incidence at any point in time. This is the number of subjects still being followed who have not had an event.
- The **number of events** is the numerator in an estimate of incidence at any point in time.
- If a subject doesn't reach the endpoint, they are **censored** at the last follow-up time.



Modeling time-to-event analysis: The Cox Model

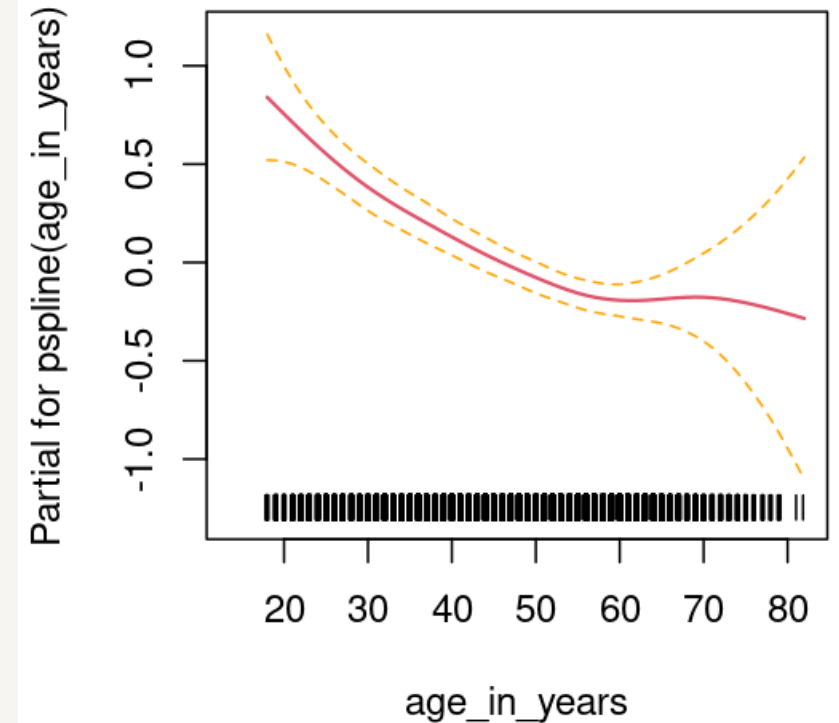
- The statistician David Cox popularized many methods in logistic regression created a strategy to incorporate time and censoring as well. This is known as the **Cox proportional hazards model**.
- In Cox modeling we model the instantaneous hazard as a function of time:

$$\lambda(t) = \lambda_0(t)\exp(X\beta)$$

- The basic assumptions are:
 - Linearity
 - Proportionality

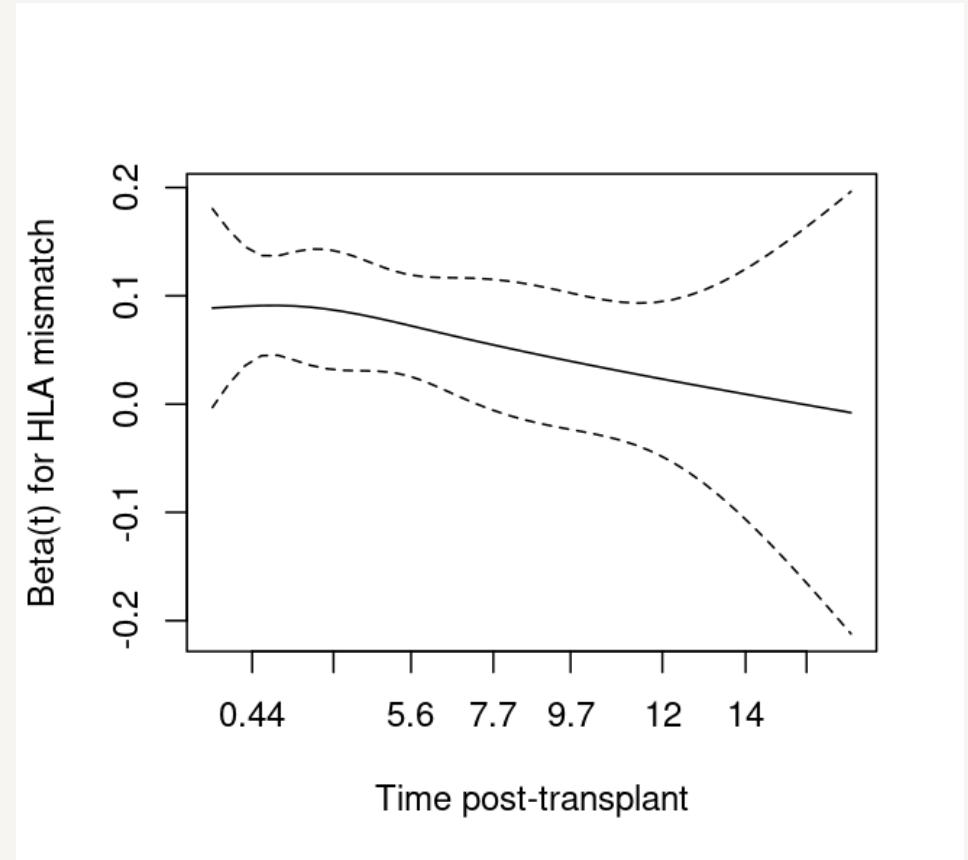
The assumption of linearity

- A frequent way to overcome a lack of linearity is to use splines.
- In many packages, you can apply a non-linear transform and check the fit.
- In R, the developers of the 'survival' package created a convenient test that directly looks at the linear portion of a spline versus the non-linear terms.
- Graphically, we can also look at the 'term plot' which is essentially a plot of predicted values over a range of predictor values.



The assumption of proportionality

- Proportionality is important from one perspective because it means that the effect of a variable does not depend on time.
 - In other words, as curves start to separate, then continue to do so at a constant rate.
- A type of residual known as the Schoenfeld residual was proposed as a test of proportionality.
 - The 'formal' test is really a test on the linearity of a spline fit through these residuals after a specific weighting.
 - Many statisticians just use the 'pencil' test.

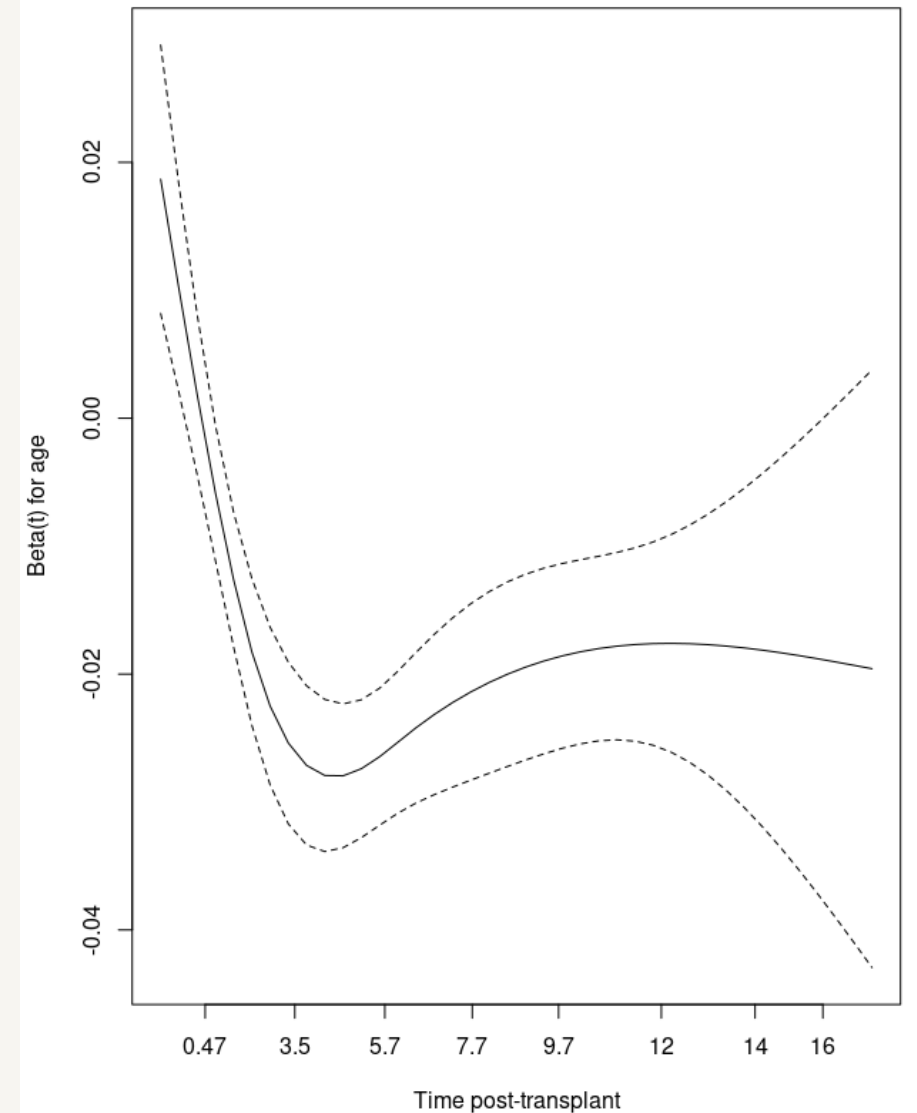


Side note about software

- In R, the 'survival' package has been a popular package for survival analysis.
 - The default in Cox modeling and Kaplan-Meier analysis is to use Efron ties which tend to be more exact than Breslow.
 - The routine 'cox.zph()' will give a test and plots of proportionality.
- In Python, the most popular package is the 'lifelines' package.
 - The default for Cox modeling is the Breslow ties which is computationally faster but less accurate.
 - The fit object has a 'check_assumptions' property that will check for proportionality.
 - The author of the lifeline package has said that proportionality testing doesn't matter (https://lifelines.readthedocs.io/en/latest/jupyter_notebooks/Proportional%20hazard%20assumption.html#Do-I-need-to-care-about-the-proportional-hazard-assumption?).

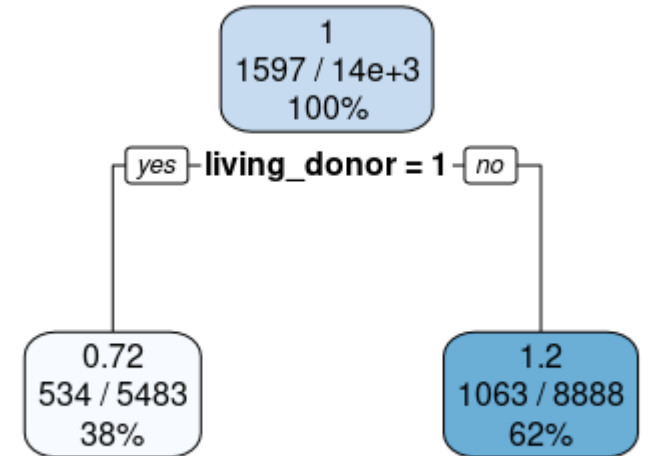
An example of non-proportionality

- The relationship between age and transplant can violate the assumption of proportionality.
- Older recipients are at higher risk of graft failure early.
- Older recipients are also at lower risk of graft failure later.
- Non-proportionality can be informative!



Machine learning alternatives: Survival trees

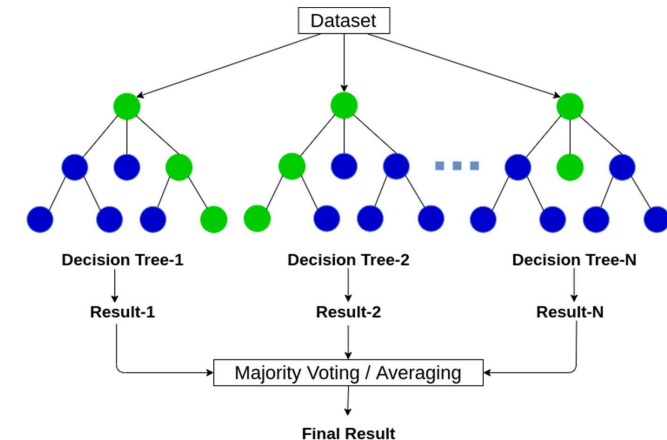
- (Survival) trees divide the feature space based on some criteria.
- If you can come up with an objective function like log likelihood, you can come up with rules to splitting patients into buckets.
- A popular method in R (rpart) assumes an exponential baseline hazard and then fits things as though it is a Poisson regression problem.



Machine learning alternatives: Random forests

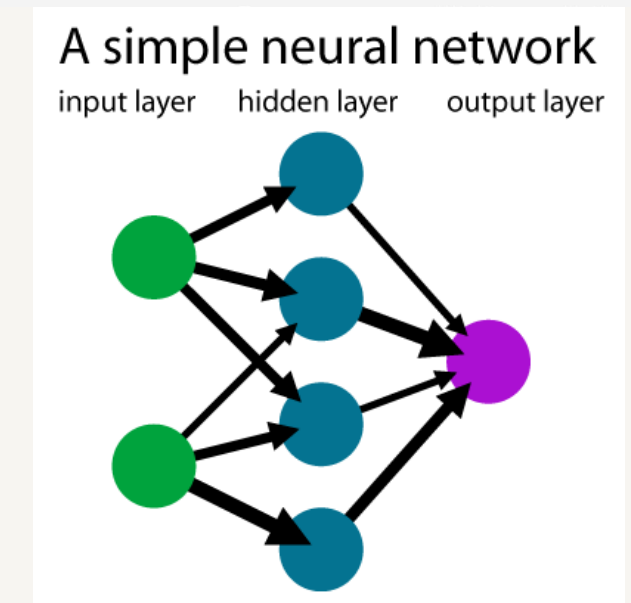
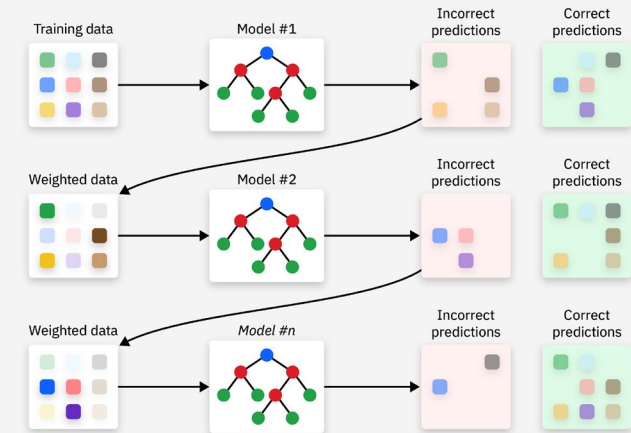
- Random forests are a series of trees which are then assembled after the fit.
- If each tree is deterministic, why fit more than one?
 - A random sample of the data is taken with each tree.
 - A random sample of the variables is taken with each tree.

Random Forest



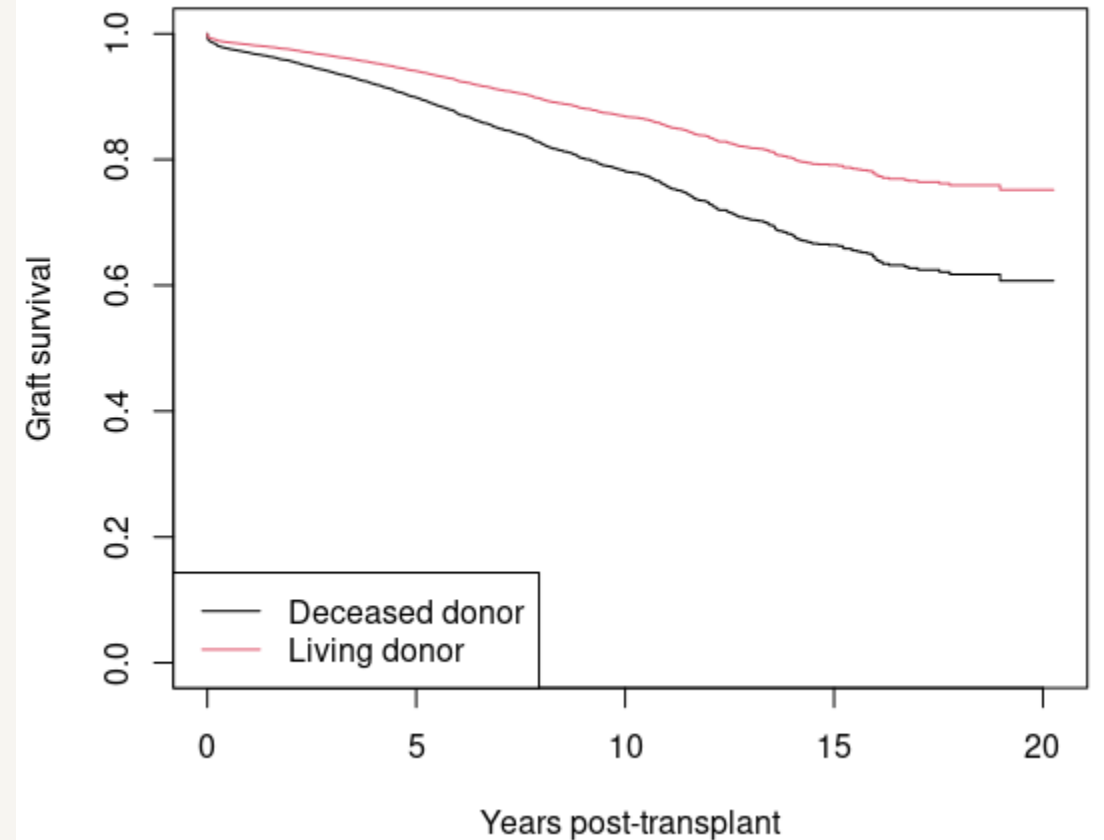
Machine learning alternatives: Others

- Instead of fitting trees in parallel (forests), we can fit trees sequentially and weight 'missed' observations (boosting).
 - The most popular methods for boosting include XGBoost and LightGBM.
- Neural network methods with a survival endpoint exist as well.
 - The partial likelihood is not standard in packages like TensorFlow/Keras or Pytorch, but there are wrapper packages.
- Support vector machines for survival exist but they don't seem to be applied very often.



Output from survival models

- Regardless of the method, the output is typically a risk of event over time.
- In Cox models, this is a combination of the non-parametric baseline hazard and the exponential of the linear predictor.
- Machine learning is more complicated and may depend on the method.



A mini rant

- Survival methods can sometimes be an after thought for people applying machine learning.
- Frequently, ML/AI users will just model a survival endpoint as a binary outcome and just drop the time. Don't do this!
- There are ways to simplify survival models.
 - Landmarks – Take all patients with a set amount of follow-up and model whether the event happened within that time point.
 - Stratify follow-up time and model the count with a Poisson model.

What about my assumptions!?

- When using these machine learning methods, the assumptions are basically already addressed.
- Trees, forests, etc. partition the feature space and create several 'local' models.
 - This overcomes non-linearity because models are formed in small intervals for each predictor.
 - This also overcomes proportionality issues because it can set different baseline hazard functions for each group.
 - This can even implicitly incorporate interactions!

Cox models vs. Machine learning

COX MODELS

- Specific functional form with a non-parametric baseline hazard and parametric relationship to predictors.
- Assumes linearity in predictors.
- Assumes constant predictor effects.

MACHINE LEARNING

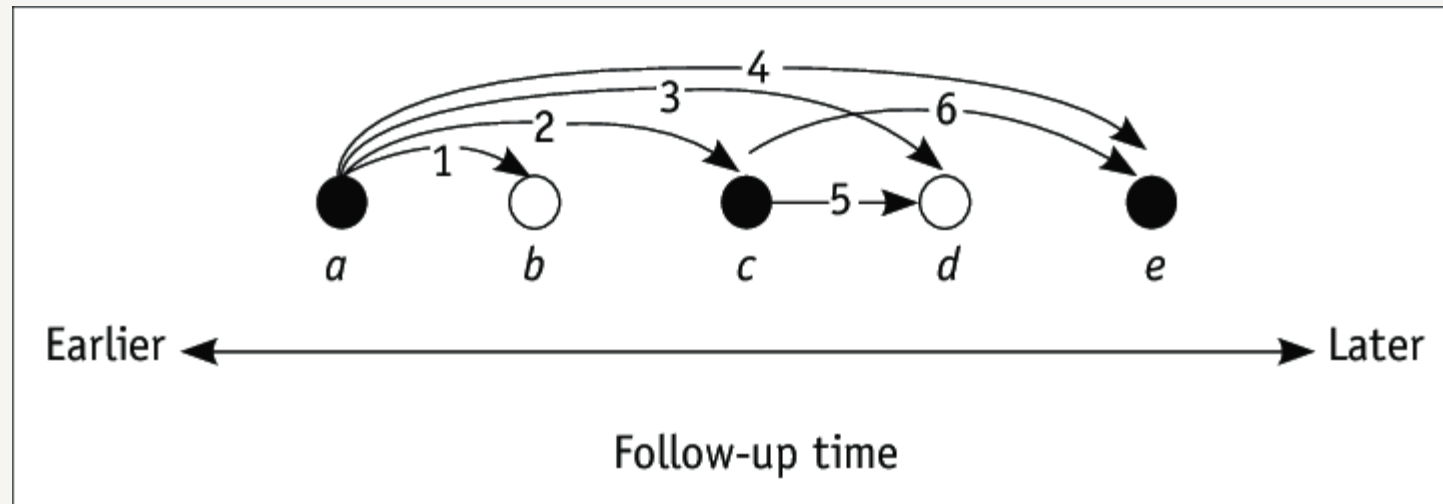
- Many hyperparameters to optimize such as minimum bucket size, split size, etc.
- Little to no assumptions depending on the method.

How do I evaluate a model?

- Suppose that I fit a machine learning model – how do I know if I did well?
 - Time-to-event is more than just an accuracy, there is a time component to it as well.
- How do I show if machine learning is justified beyond a simpler Cox model?
 - If we come up with a metric of performance, I need to be able to calculate it on the output from a machine learning model as well as the Cox model.

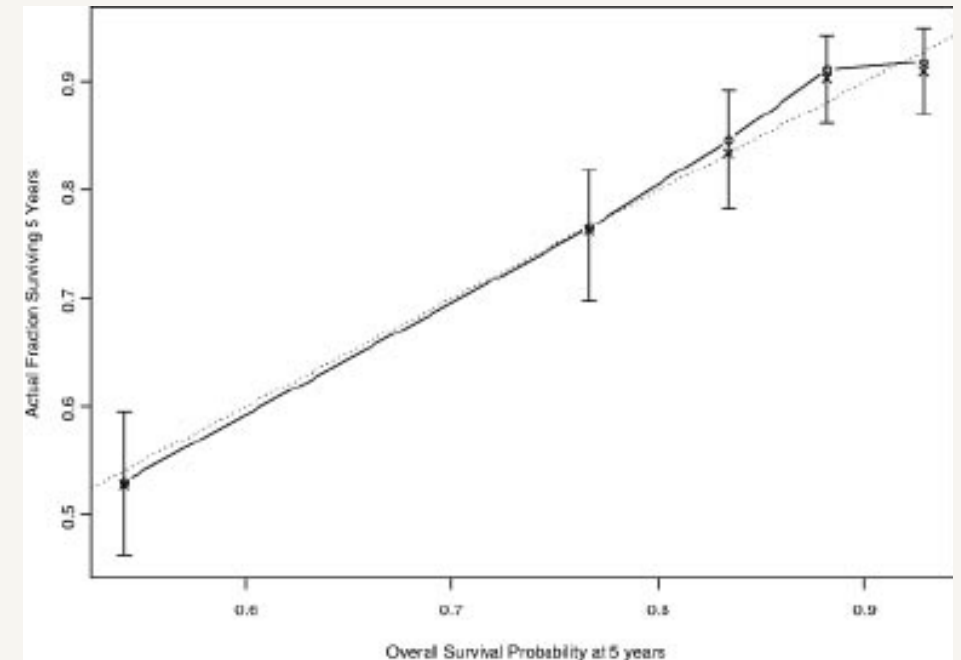
Concordance

- Take a pair of observations and predict the risk of event for each observation.
 - The pair is **concordant** if the model predicts which event happened first.
 - The pair is **discordant** if the model predicts it backwards.
 - If both observations are censored, drop the pair.
 - If one of the two doesn't meet the event, concordant status can be determined from the timing.
- The **concordance** is then the proportion of pairs observed to be concordant.



Calibration

- Importantly, concordance isn't a measure of how concordant or discordant a model is.
 - If we predict a pair as 98% vs. 5% for risk we get the same concordance as 49% vs 48%.
- Calibration instead looks at the rate of events within predicted risk groups.
 - Take a group of risks by year 5 ranging from 0-5%. The proportion of events in that group should be 0-5% by year 5.



Comparing Cox models with machine learning

- The most typical way to compare multiple models is by choosing one or more metrics of performance (concordance, calibration, etc.) and calculating these on one or more validation datasets.
- There are a lot of different validation methods such as k-fold cross-validation, bootstrapping, single train/test splits, and external validations.
 - The most important thing is that each aspect of your model building process does not touch the datasets that you are using for testing.

An aside: Wifi chips and the Internet of things

- In the late 2000s changes in internet protocols, cloud storage, and cheap wi-fi enabled chips connected devices.



vulnerable to hacking.

and obsolescence.

to repair.

cerns with how our devices use c

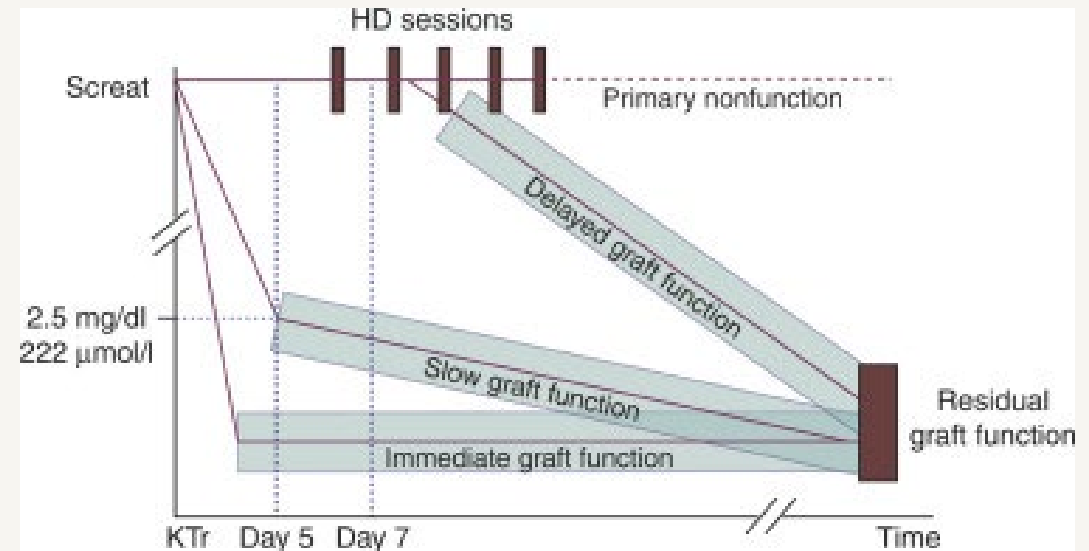


Implementing a machine learning calculator in the EHR

- In order to get practitioners to use a predictive model, it should be 'within reach' by placing it in the electronic health record (EHR).
 - Doing so is not trivial.
- I was brought in after a model was built in order to have the model predictions output directly in the electronic health record.
- A boosted tree model (catboost) was built to predict an ordinal outcome. An important note is that catboost is a boutique package/model which means the package and all dependencies need to be controlled when using it.

Delayed graft function

- When a kidney is less than ideal, there is a chance that it may not work immediately after transplant.
- Patients with delayed graft function will need to remain on routine dialysis until the kidney begins to work.
- Typically delayed graft function is a binary variable defined as the need for dialysis within the first week post-transplant.

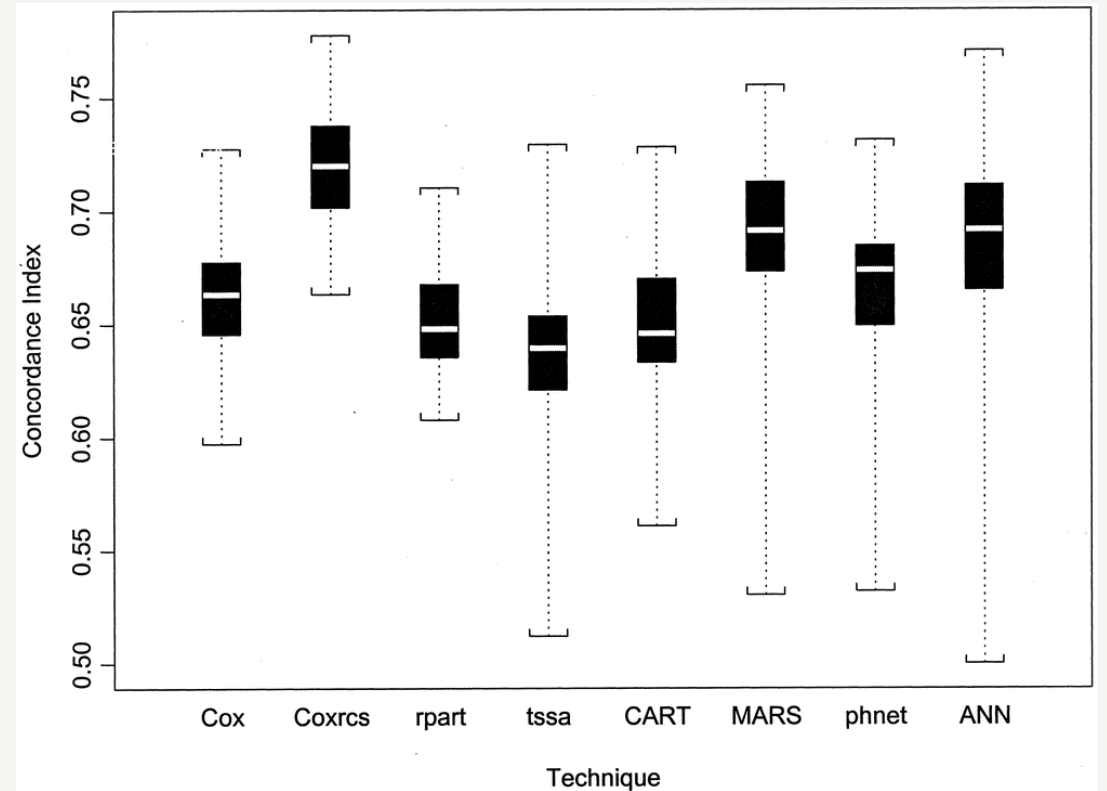


Implementing a machine learning calculator in the EHR

- To incorporate an ordinal logistic prediction (1 month timeline):
 - Calculate the predicted probability and class within SQL.
 - Output.
- To incorporate a machine learning model (9 month timeline):
 - Create code within python. This will implement a proprietary Epic package.
 - Using a Google virtual machine:
 - Create a Slate Docker instance
 - Package code with required libraries in a version-controlled way
 - Zip everything together to deliver to IT team
 - Troubleshoot errors iteratively.

Kattan et al 2003

- In 2003, Michael Kattan compared relatively new methods in survival with Cox models across multiple datasets (DOI: 10.1097/01.ju.0000094764.56269.2d).
- He found that while machine learning often outperformed normal Cox models, once non-linearity was incorporated using restricted cubic splines, the Cox model will out-perform machine learning methods.

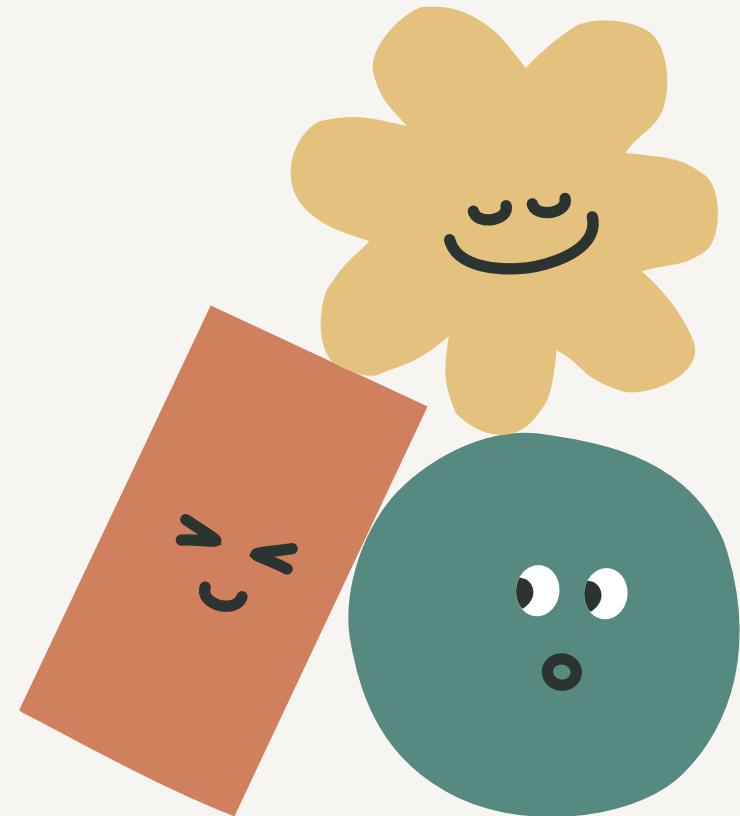


A different use for machine learning

- Machine learning can be a very effective way of testing assumptions – if machine learning substantially improves model performance then maybe your model is mis-specified.
- I propose a new strategy:
 1. Fit a Cox model
 2. Fit a machine learning model
 3. Compare them on some metric of performance
 4. Re-evaluate the Cox model if machine learning out-performs the Cox model.

“(AI) workslop is work that masquerades as good work but lacks the substance to meaningfully advance a given task.”
Furthermore, “its insidious effect is transferring effort from creator to receiver.”

BetterUp Labs in collaboration with Stanford social media lab on the Brene Brown/Adam Grant podcast.



Conclusions

- Machine learning methods exist that can circumvent the assumptions of standard survival models.
- Comparisons of simpler models with machine learning models can reveal holes in your modeling methods such as violations of proportionality or model misspecification.
- If machine learning doesn't actually show a benefit over simpler models, it could slow down projects and clog the system.

The background is a solid light green color. It features several large, semi-transparent geometric shapes: a large triangle on the left side, a large circle in the lower-left quadrant, and a large, stylized flower shape on the right side. The word "Questions?" is centered in the middle of the image.

Questions?