Development and Evaluation of Data-Driven Models using High Frequency Time Series

Anthony Wertz
Research Analyst
Auton Lab
Carnegie Mellon University

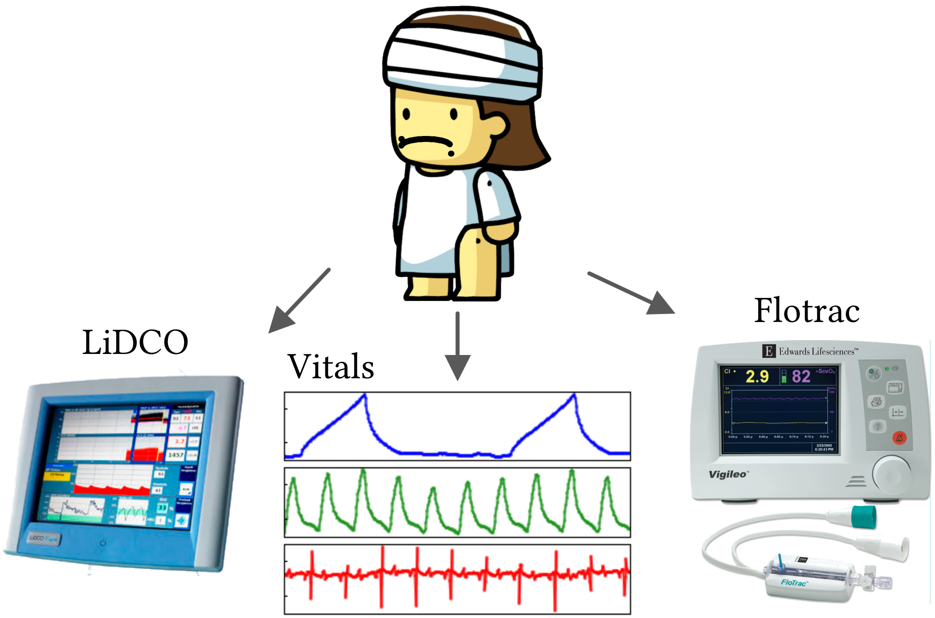
26 April 2018

Motivation

- Patient care can benefit from knowledge of patient state and disease progression.

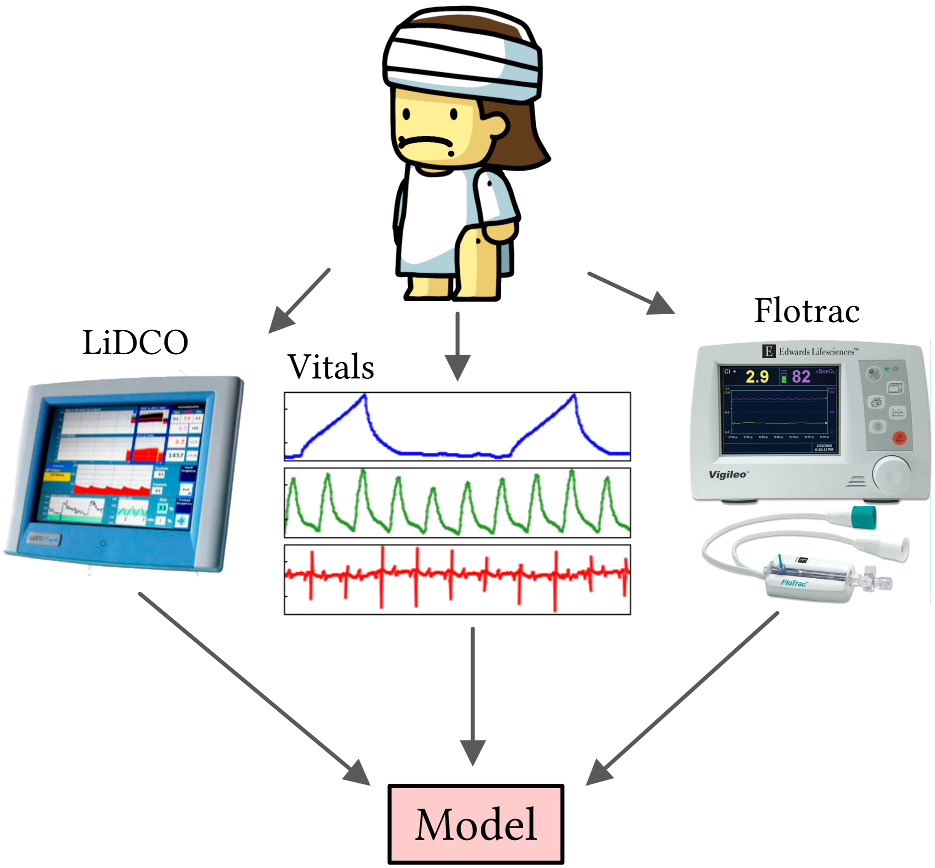
Motivation

- Patient care can benefit from knowledge of patient state and disease progression.
- Monitoring systems can help…



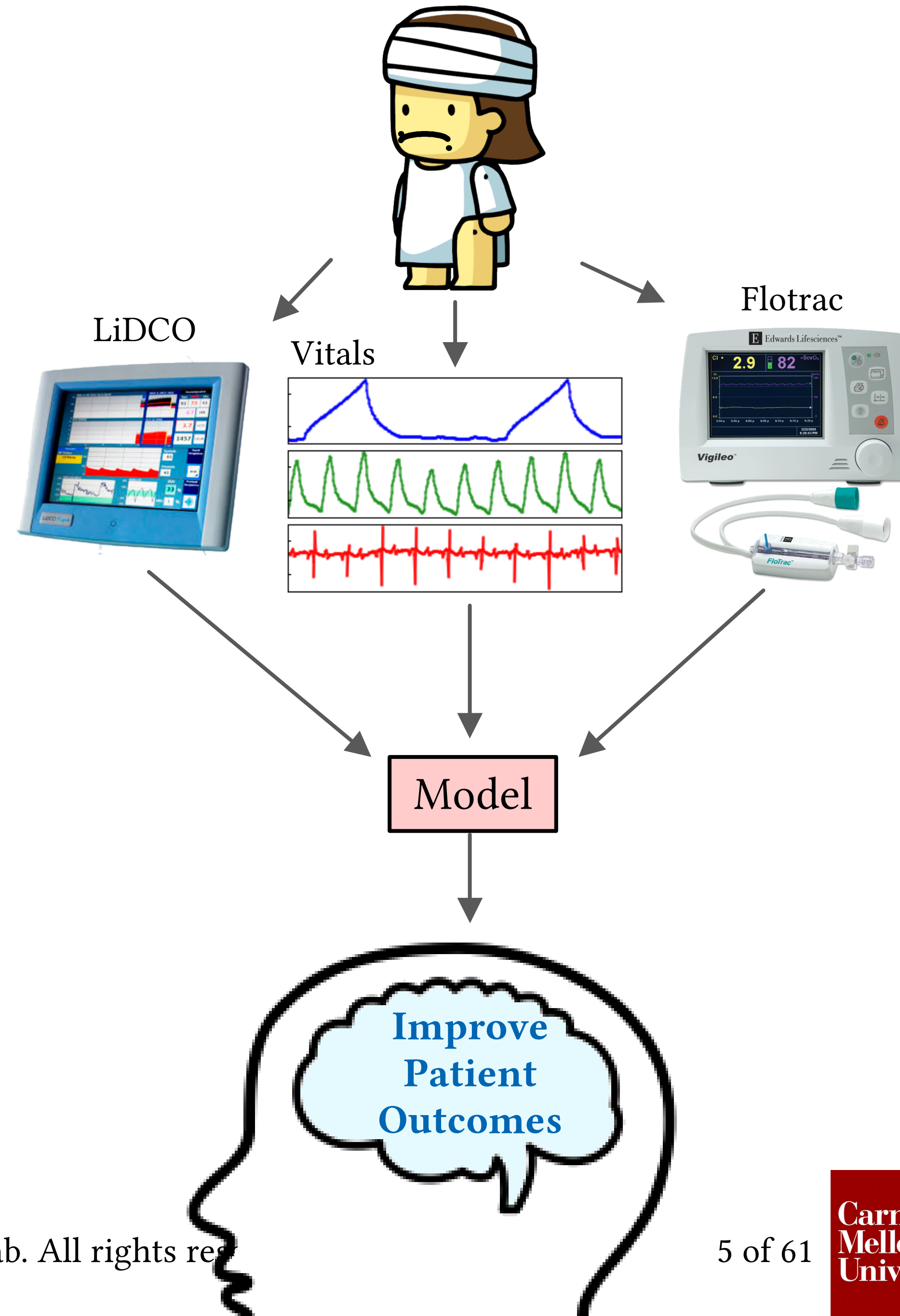
Motivation

- Patient care can benefit from knowledge of patient state and disease progression.
- Monitoring systems can help…
- …but we need models to really describe them.



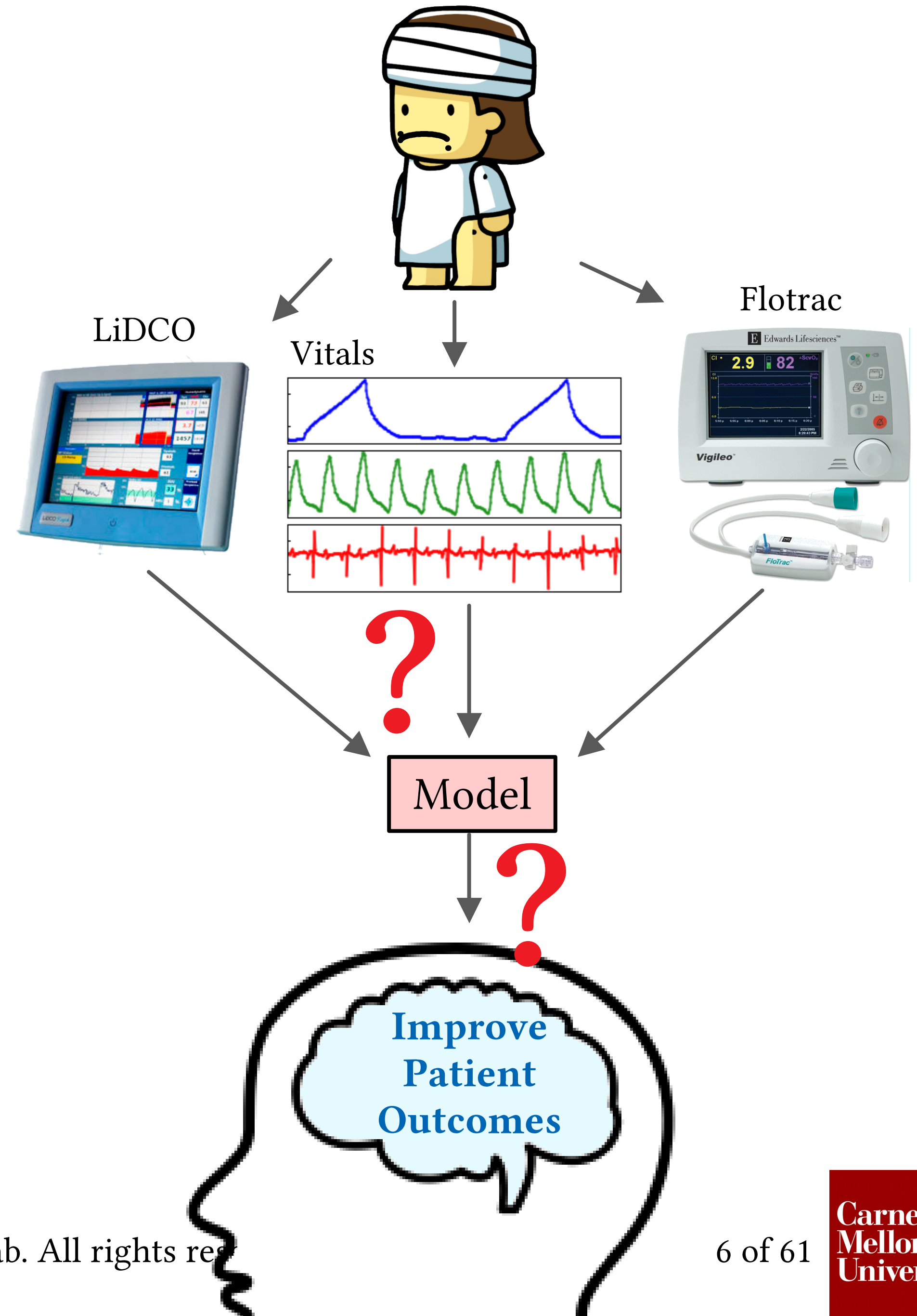
Motivation

- Patient care can benefit from knowledge of patient state and disease progression.
- Monitoring systems can help…
- …but we need models to really describe them.
- Alone, models aren't very intelligent, but we can evaluate our models to determine how to use them intelligently.



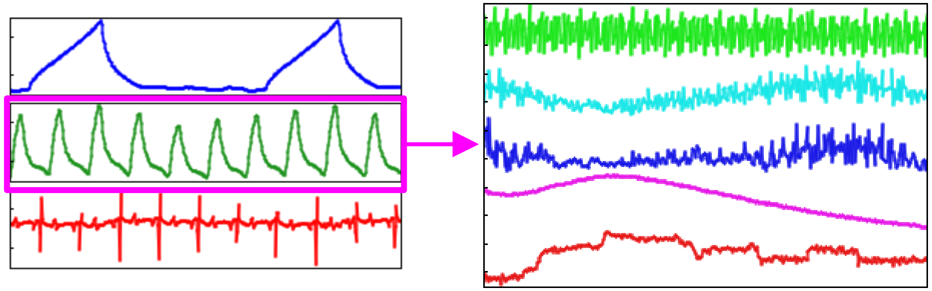
Motivation

- Patient care can benefit from knowledge of patient state and disease progression.
- Monitoring systems can help…
- …but we need models to really describe them.
- Alone, models aren't very intelligent, but we can evaluate our models to determine how to use them intelligently.
- How can we use high density data collected from patients in research and in practice?



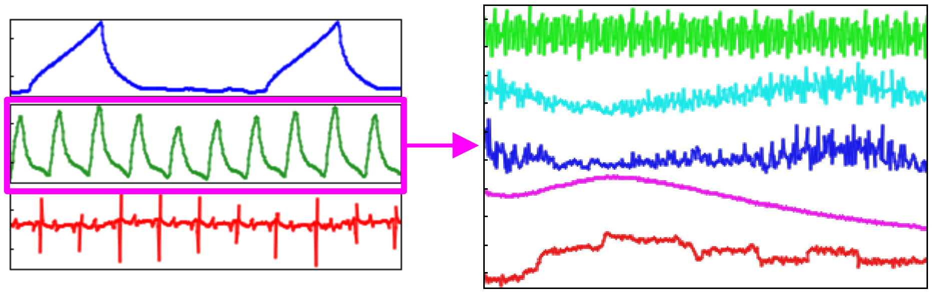
Computational Experimental Design

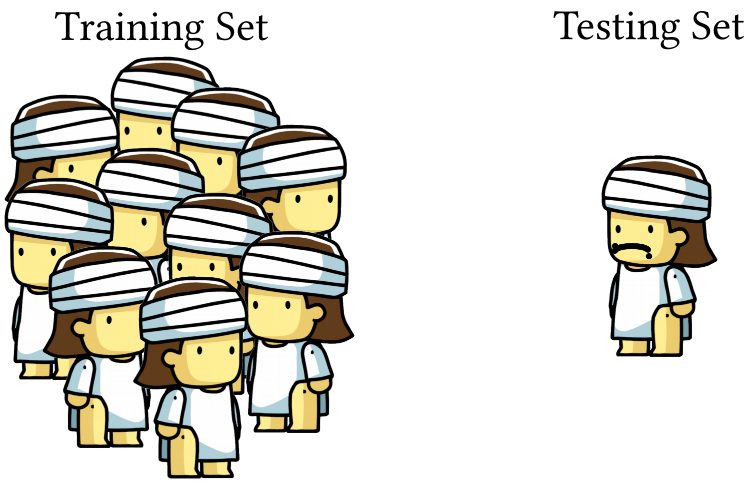
- **Featurization**: Pull out information that might be difficult for a model to discover automatically.



Computational Experimental Design

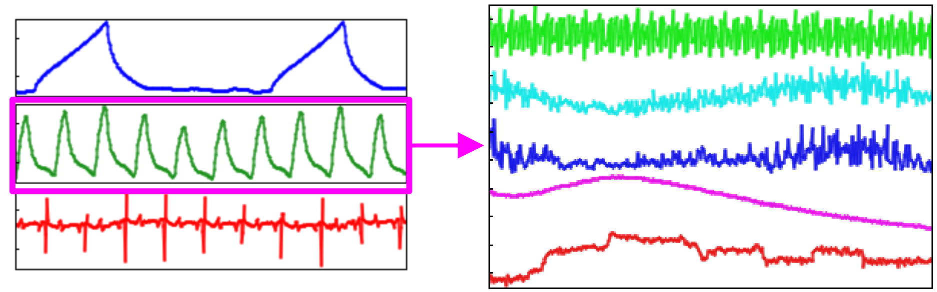
- **Featurization**: Pull out information that might be difficult for a model to discover automatically.
- **Training and Validation**: Build a good model.



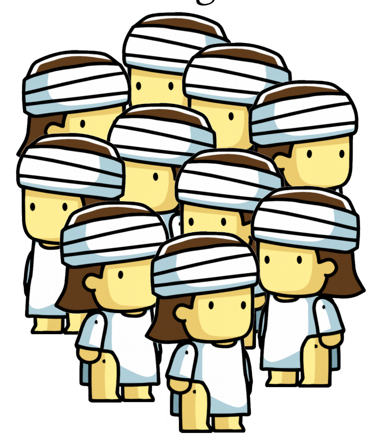


Computational Experimental Design

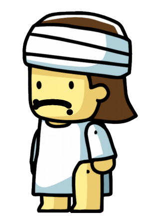
- **Featurization**: Pull out information that might be difficult for a model to discover automatically.
- **Training and Validation**: Build a good model.
- **Evaluation**: Understand the model's performance.



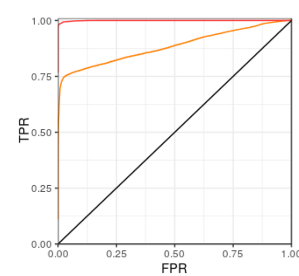
Training Set



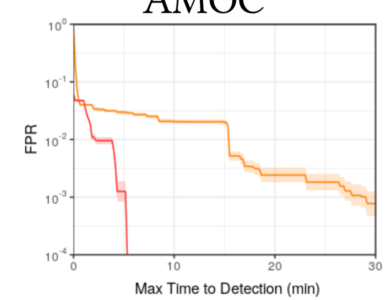
Testing Set



ROC

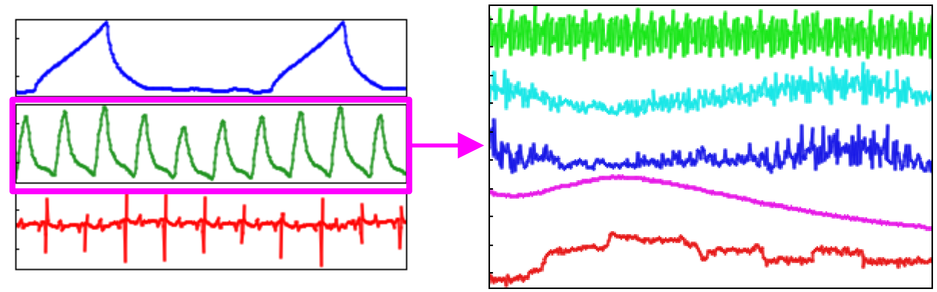


AMOC

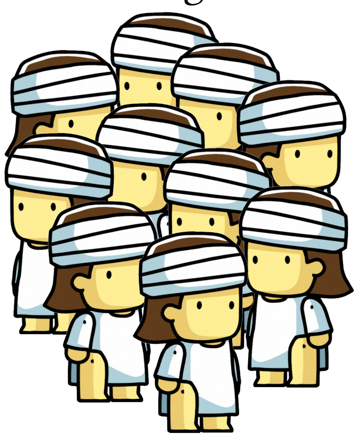


Computational Experimental Design

- **Featurization**: Pull out information that might be difficult for a model to discover automatically.
- **Training and Validation**: Build a good model.
- **Evaluation**: Understand the model's performance.
- **Operationalize**: (Optional) Use the model in a clinical setting.

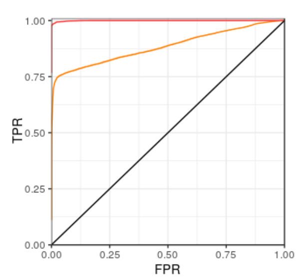


Training Set

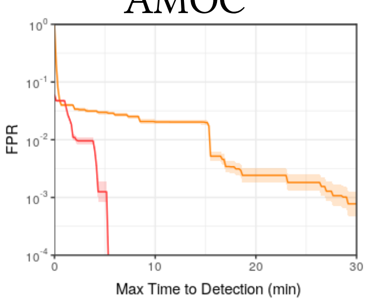


Testing Set

ROC



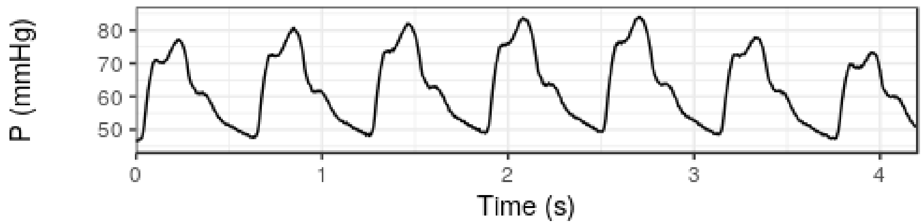
AMOC



Featurization

Time Series Featurization

- Monitoring devices can produce high density time signals. How do we analyze them?

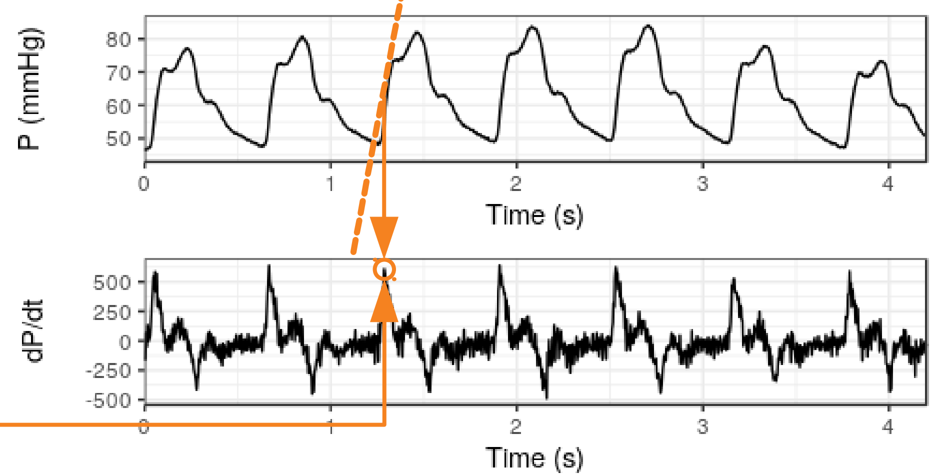


Time Series Featurization

- Monitoring devices can produce high density time signals. How do we analyze them?
- We can use instantaneous values.

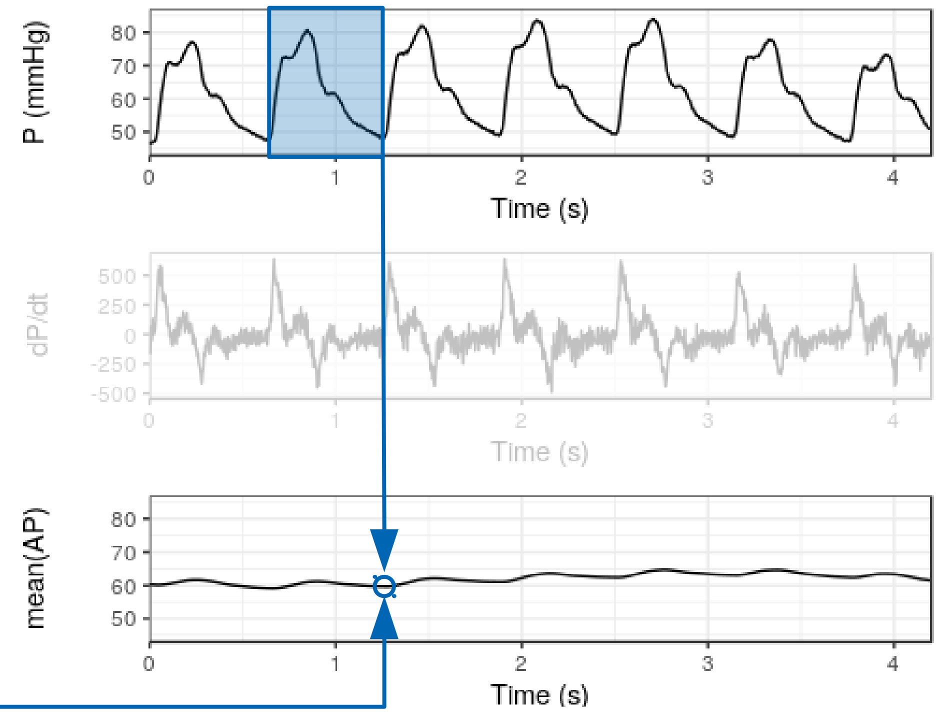
Time Series Featurization

- Monitoring devices can produce high density time signals. How do we analyze them?
- We can use instantaneous values.
- We can look at integrals and derivatives.



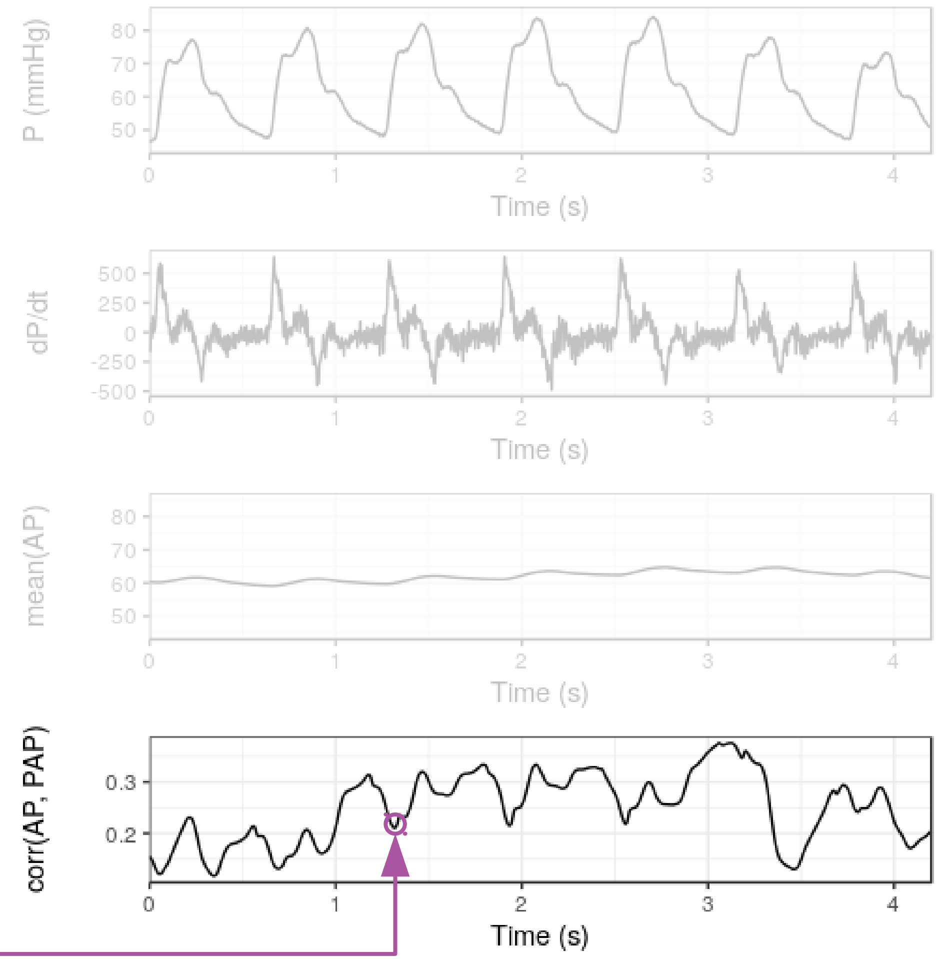
Time Series Featurization

- Monitoring devices can produce high density time signals. How do we analyze them?
- We can use instantaneous values.
- We can look at integrals and derivatives.
- We can compute features in a sliding window (statistics, trend lines, test statistics, ...).



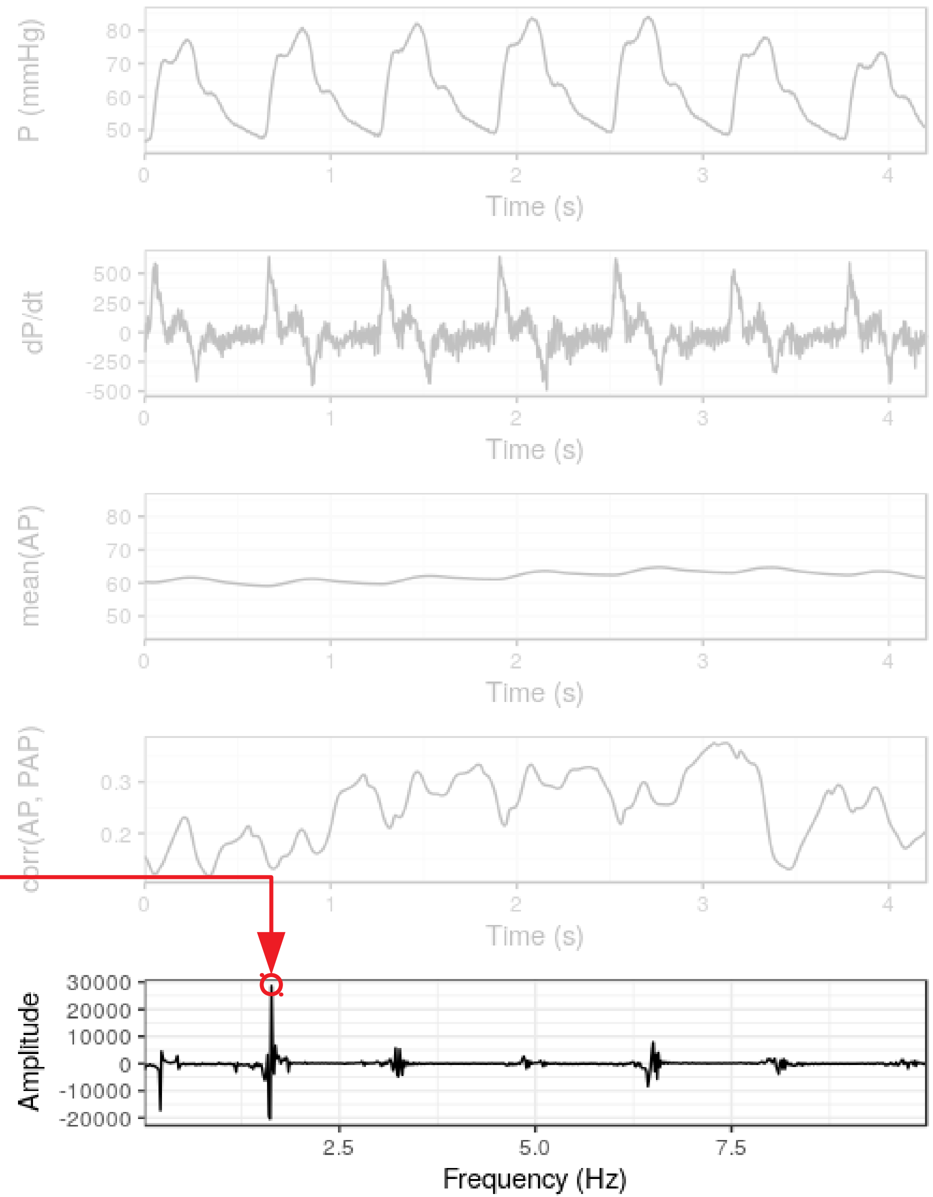
Time Series Featurization

- Monitoring devices can produce high density time signals. How do we analyze them?
- We can use instantaneous values.
- We can look at integrals and derivatives.
- We can compute features in a sliding window (statistics, trend lines, test statistics, ...).
- We can look at signal correlations, and apply all of the above techniques (e.g. rolling correlation).



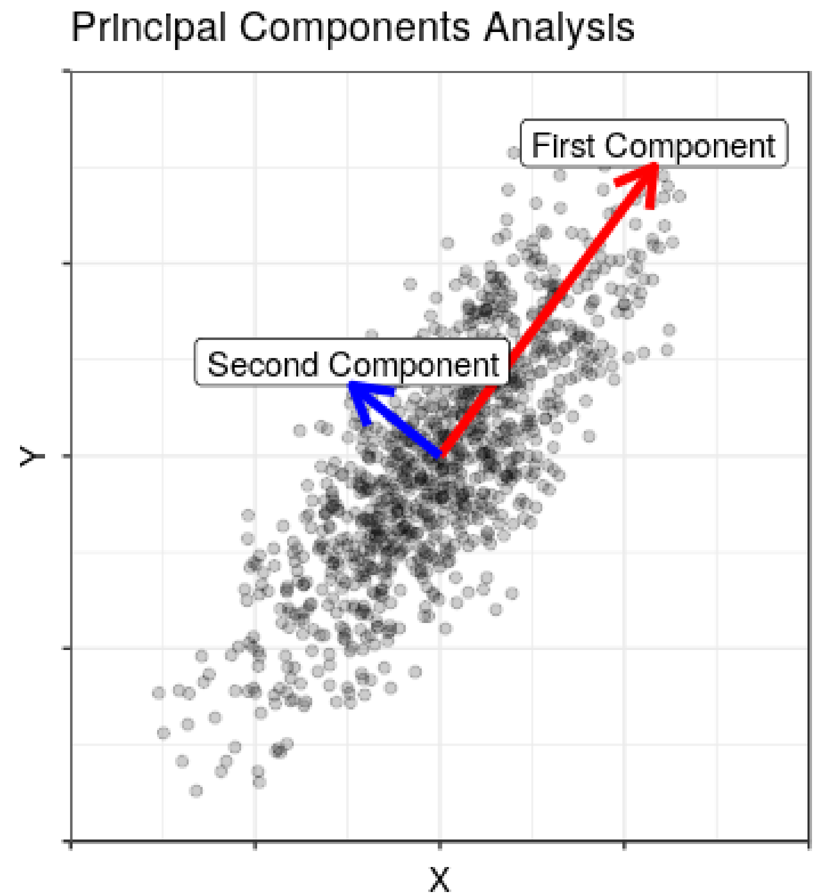
Time Series Featurization

- Monitoring devices can produce high density time signals. How do we analyze them?
- We can use instantaneous values.
- We can look at integrals and derivatives.
- We can compute features in a sliding window (statistics, trend lines, test statistics, ...).
- We can look at signal correlations, and apply all of the above techniques (e.g. rolling correlation).
- We can extract frequency components (Fourier transform, wavelet, spectral power, ...).



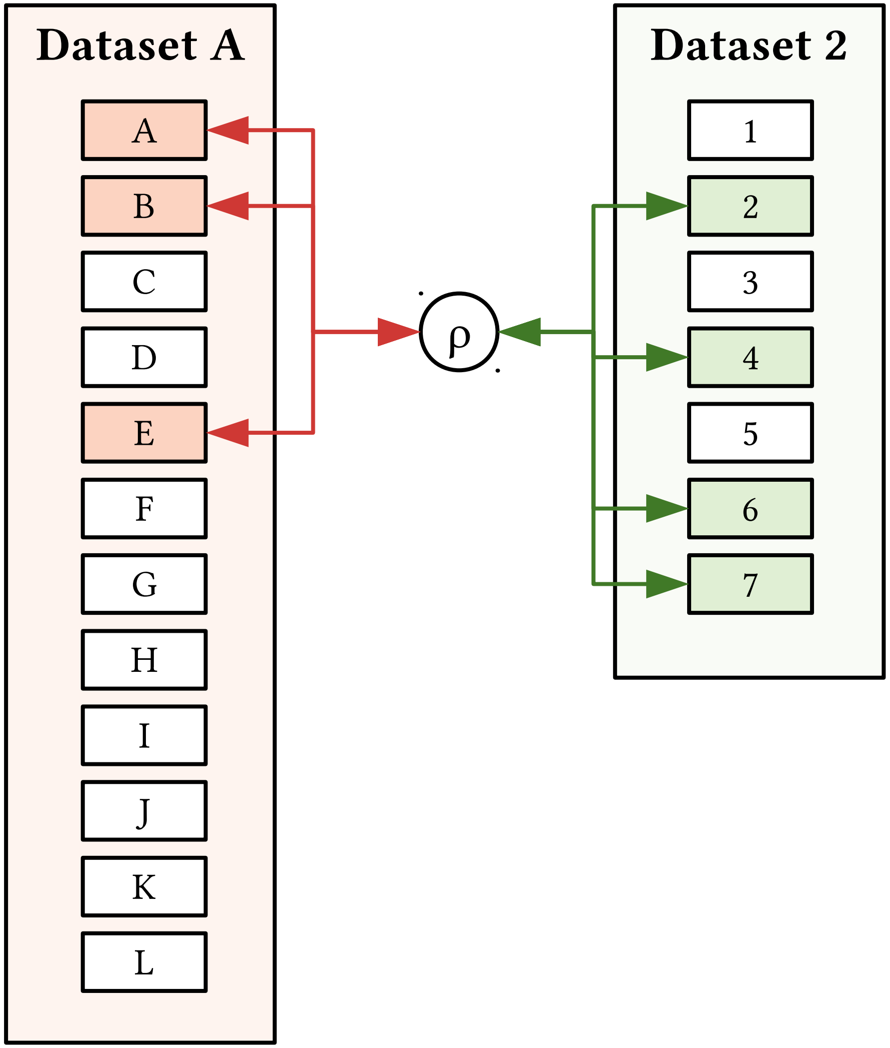
Structure of Variance in Data

- Principal Components Analysis (PCA)
  - Which correlations explain the most variation in the data?
  - Dimensionality reduction.
  - Anomaly detection.



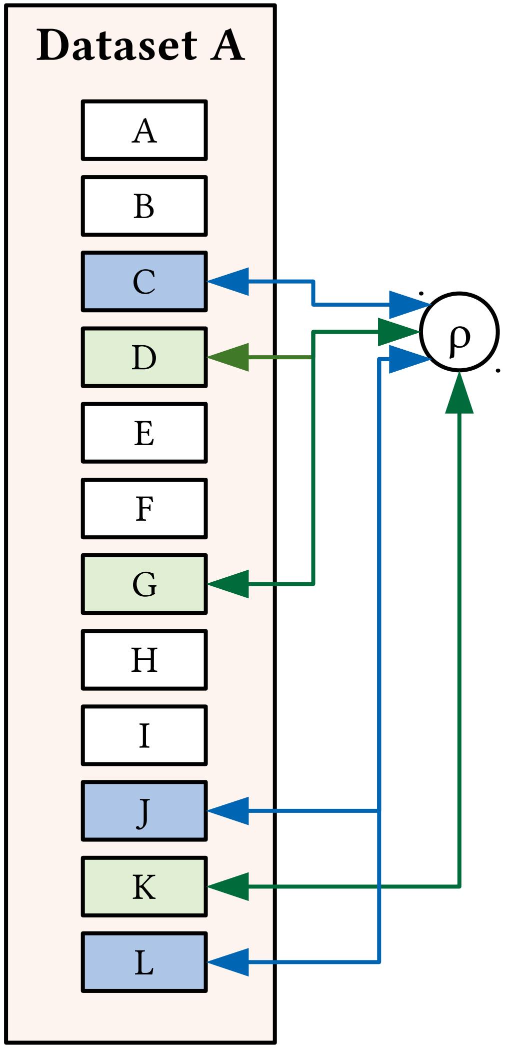
Structure of Variance Across Datasets

- Principal Components Analysis (PCA)
  - Which correlations explain the most variation in the data?
  - Dimensionality reduction.
  - Anomaly detection.
- Canonical Correlation Analysis (CCA)
  - Which correlations between features of two datasets explain the most variation in the data?



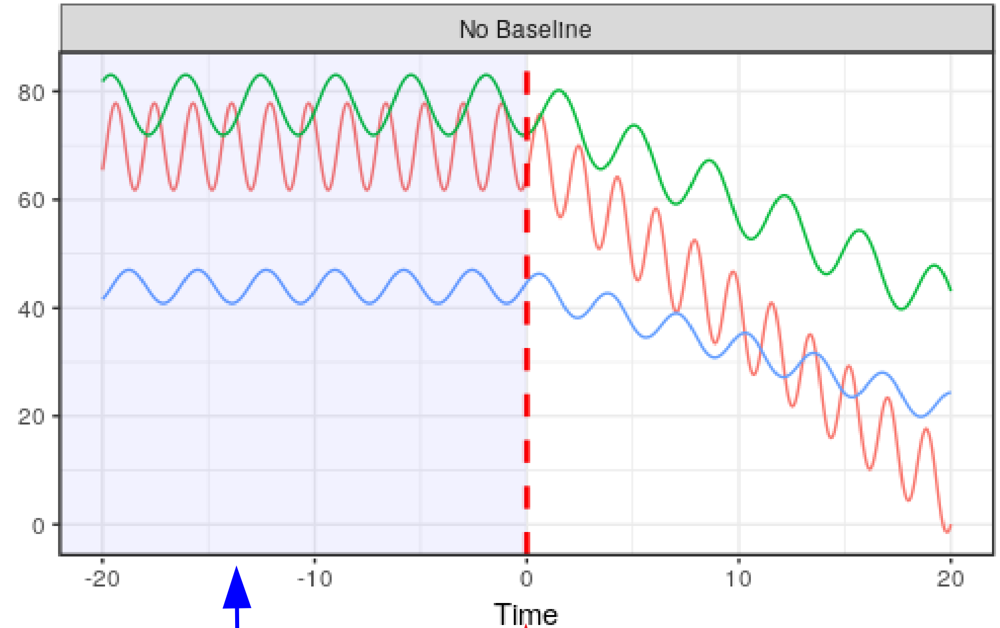
Structure of Variance Across Features Subsets in a Single Dataset

- Principal Components Analysis (PCA)
  - Which correlations explain the most variation in the data?
  - Dimensionality reduction.
  - Anomaly detection.
- Canonical Correlation Analysis (CCA)
  - Which correlations between features of two datasets explain the most variation in the data?
- Canonical Autocorrelation Analysis (CAA)
  - Which correlations between subsets of features in a single dataset explain the most variation in the data?



Significant Variability may be Seen in Patient Vitals

- Patients can be very different when stable.

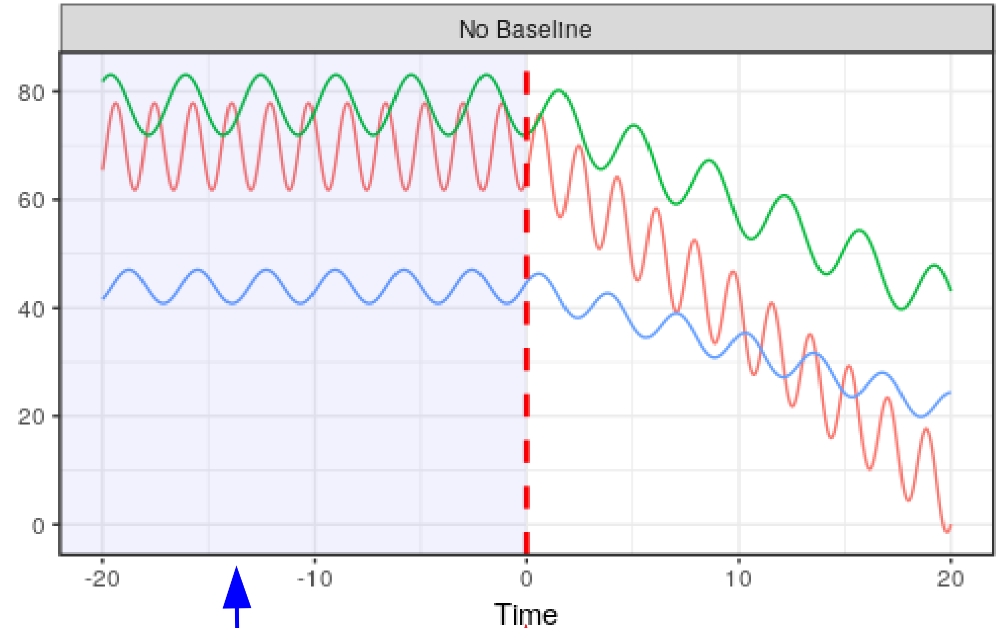


Stable before some event.

After time t=0, something interesting starts to happen to the signals.

Difficult to Set Good Detection Threshold

- Patients can be very different when stable.
- What **threshold** yields fast detection of event at t=0 and few false alarms for all patients?

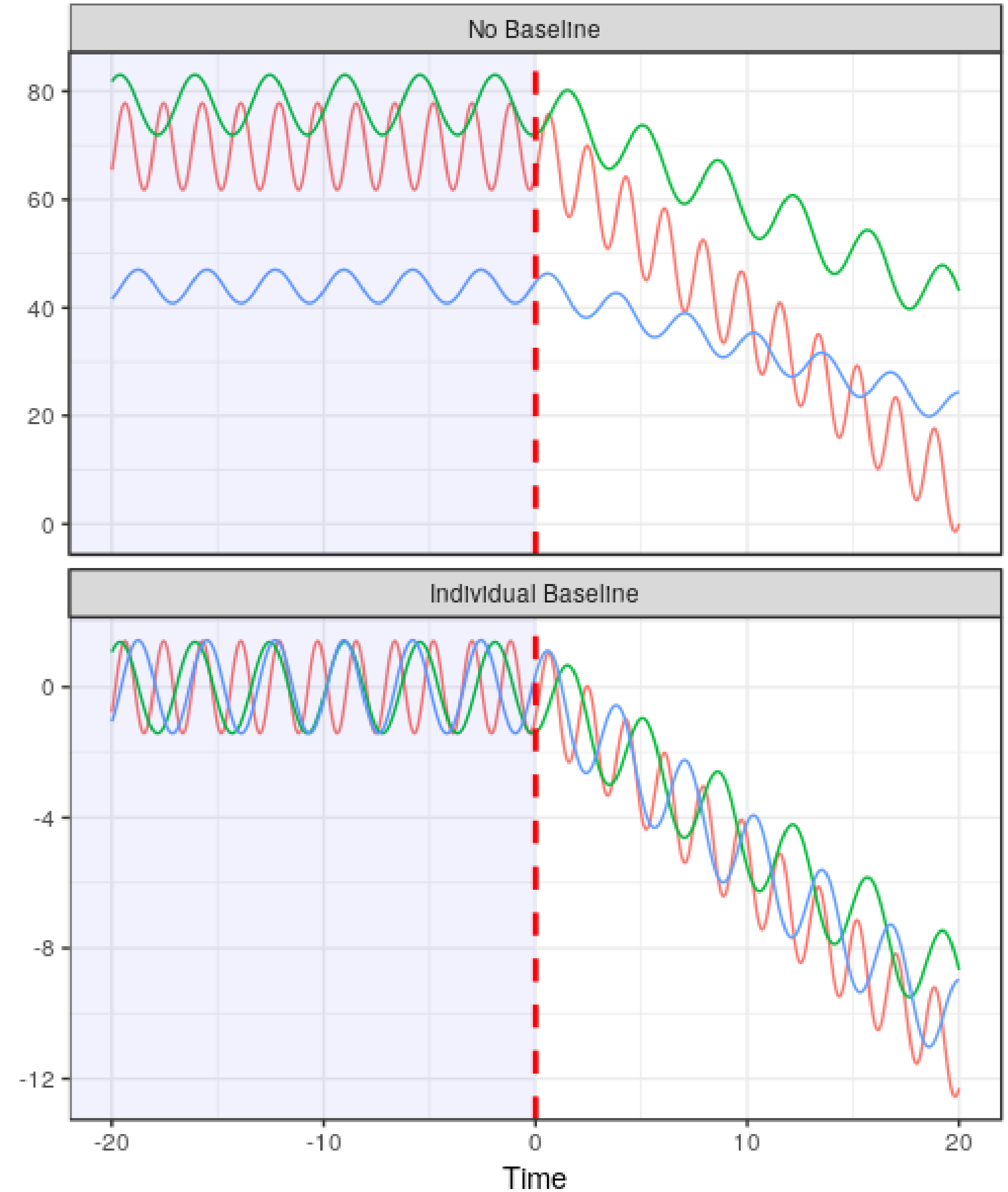


Stable before some event.

After time t=0, something interesting starts to happen to the signals.

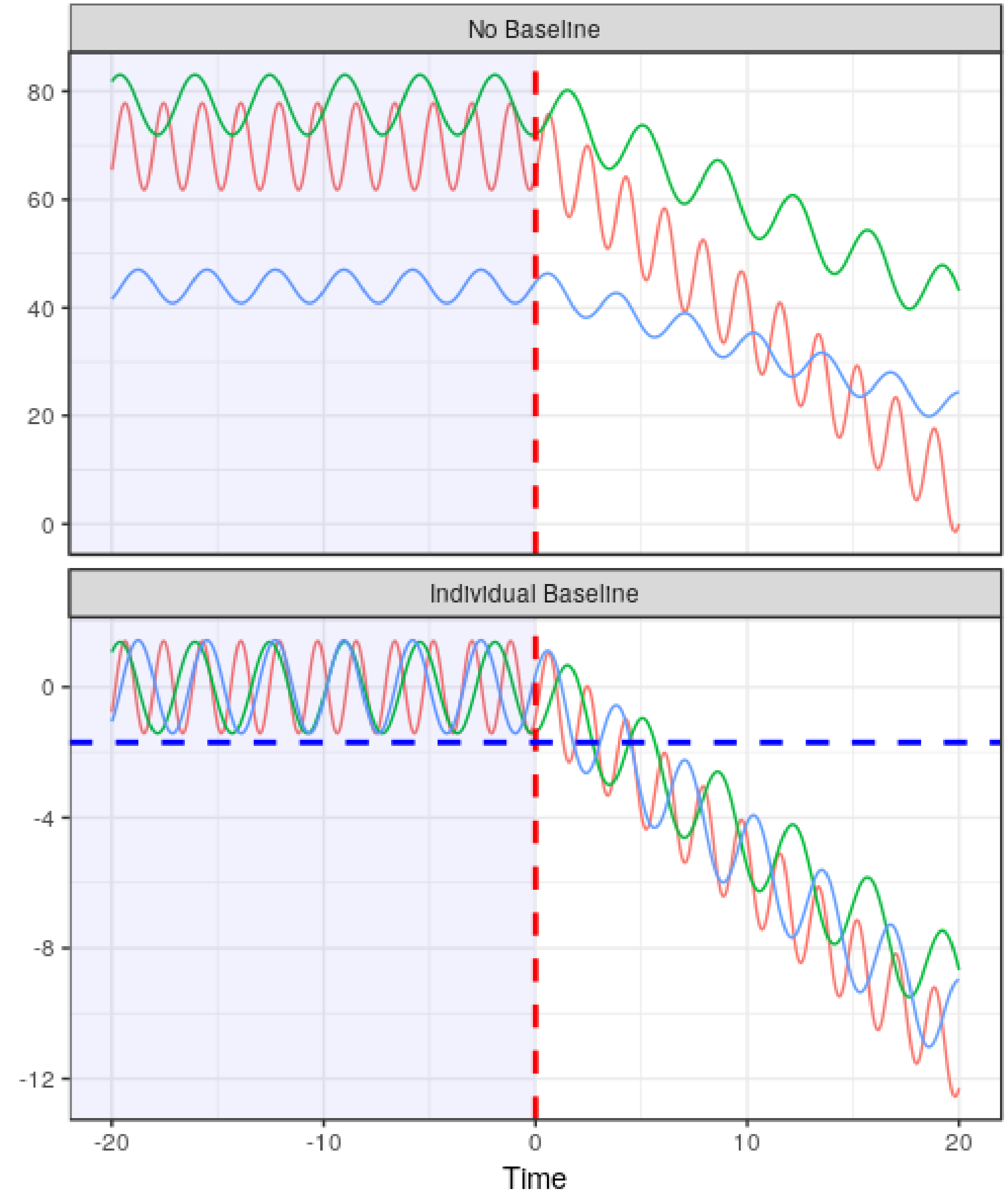
Personalized Normalization Reduces Variation

- Patients can be very different when stable.
- What threshold yields fast detection of event at t=0 and few false alarms for all patients?
- Assume some regularity in the baseline period:
  - Center on the mean.
  - Scale by its standard deviation.



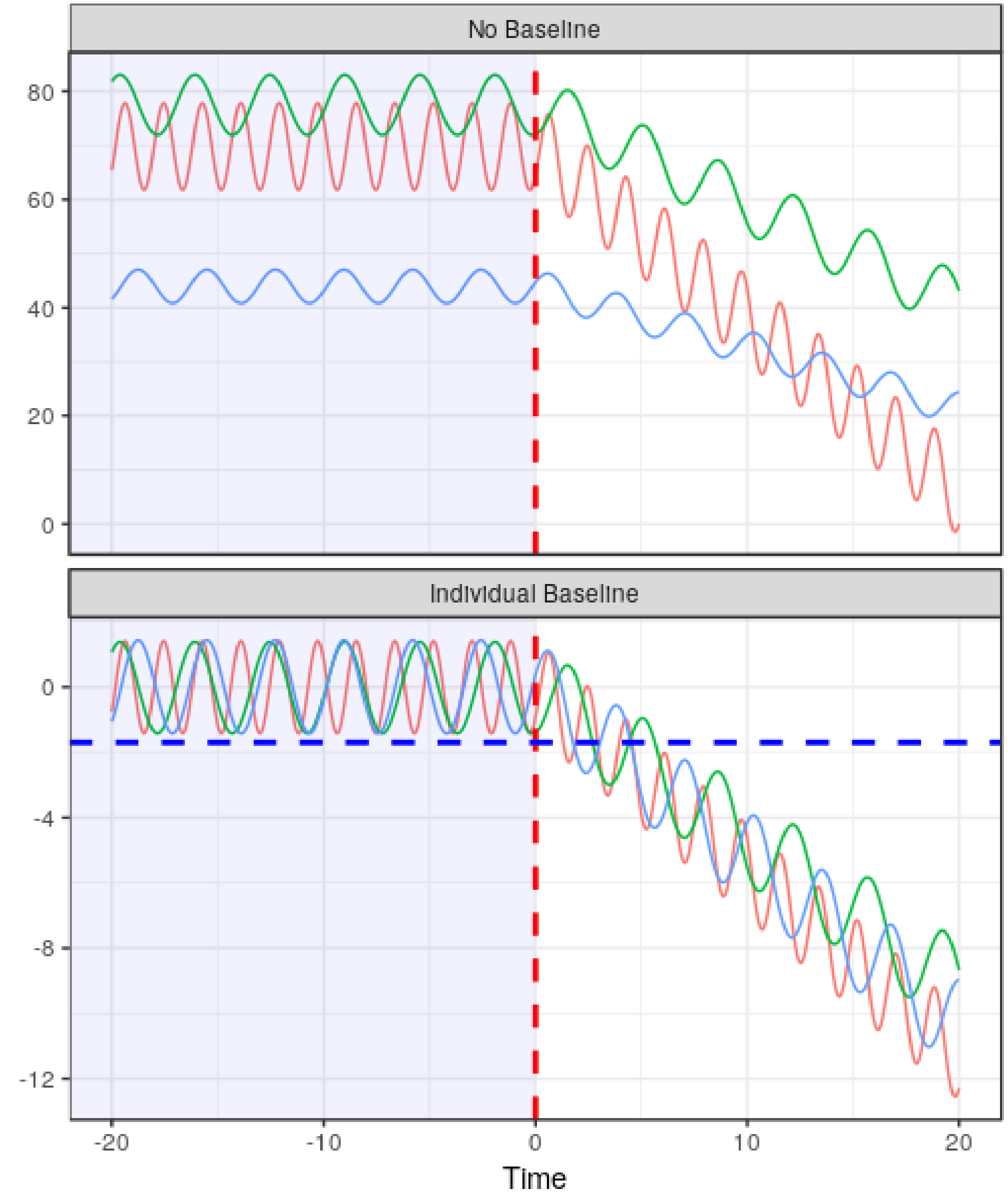
Detection Threshold can be Set After Normalization

- Patients can be very different when stable.
- What threshold yields fast detection of event at t=0 and few false alarms for all patients?
- Assume some regularity in the baseline period:
  - Center on the mean.
  - Scale by its standard deviation.
  - Now we can find a threshold for this data the yields fast detections and few false positives.



Personalized Normalization has Caveats

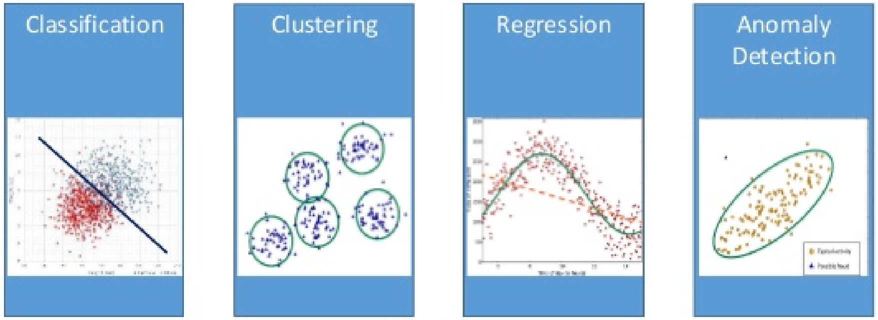
- Patients can be very different when stable.
- What threshold yields fast detection of event at t=0 and few false alarms for all patients?
- Assume some regularity in the baseline period:
  - Center on the mean.
  - Scale by its standard deviation.
  - Now we can find a threshold for this data the yields fast detections and few false positives.
- For this to work we need to collect data when we know the patient is stable.
  - Not available for every patient.
  - But can be captured for patients prior to, for example, surgery.



Model Training and Validation

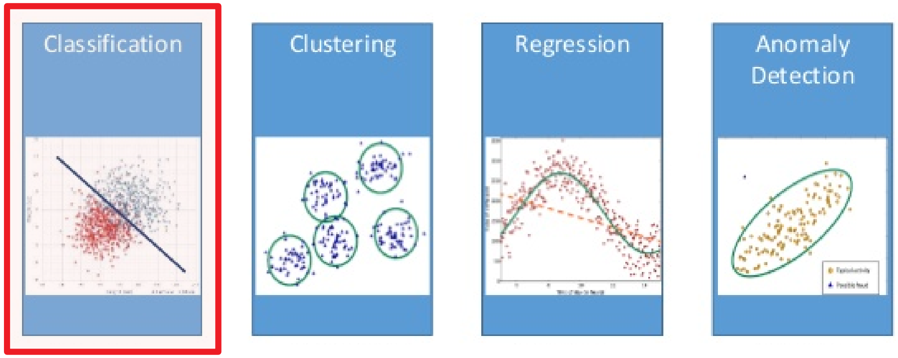
Algorithm Selection

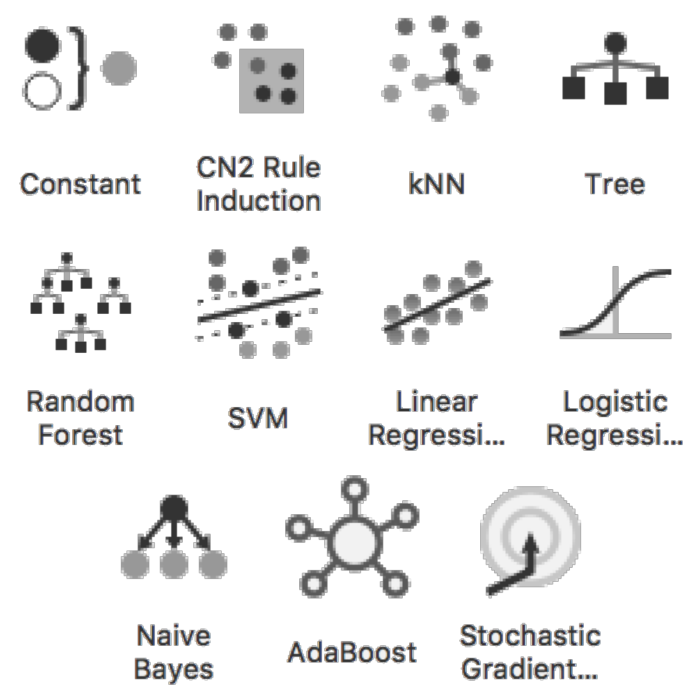
- Which algorithm will depend on what type of task:
  - Classification?
  - Clustering?
  - Regression?
  - Anomaly detection?



Many Classification Models to Choose From

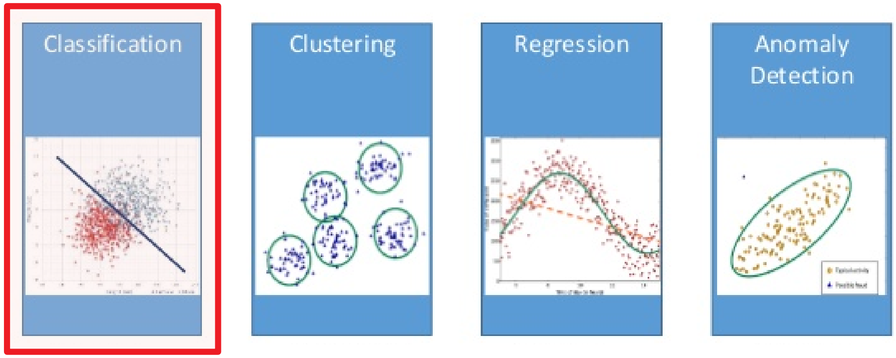
- Which algorithm will depend on what type of task:
  - Classification?
  - Clustering?
  - Regression?
  - Anomaly detection?
- We'll focus on building *classifiers.*
  - Training and validation is largely the same between types.
  - Evaluation will change.

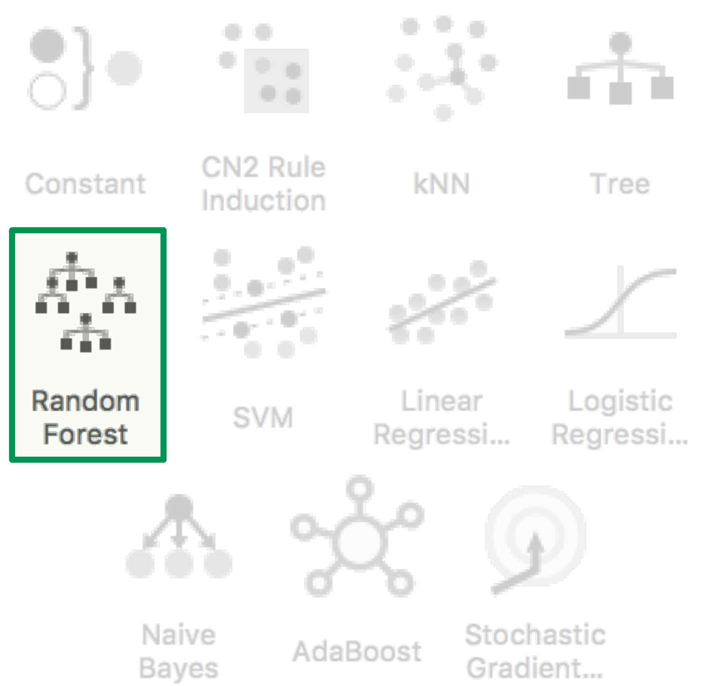




Random Forest is a Good Start

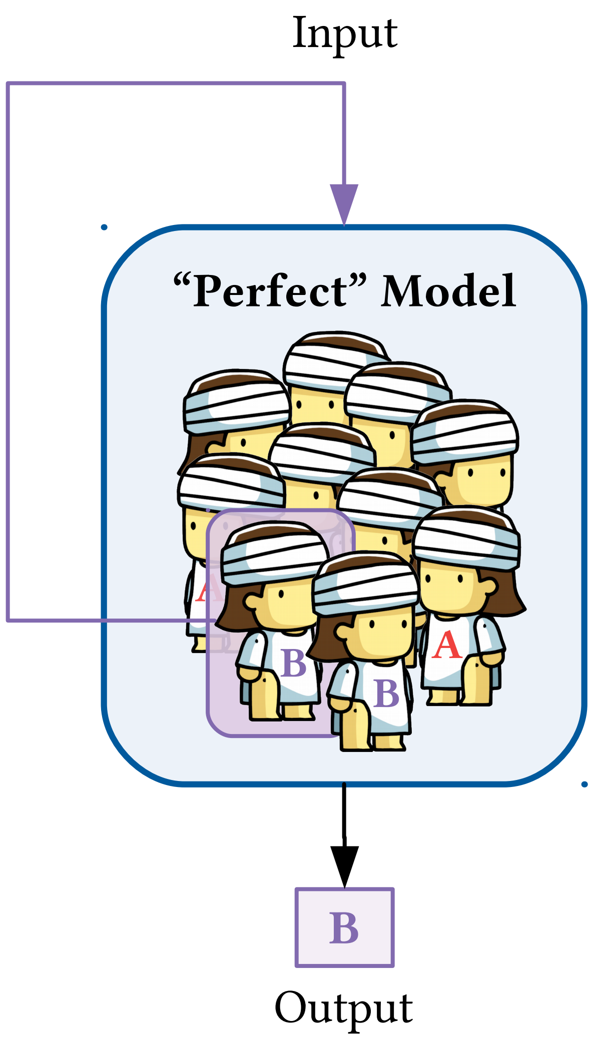
- Which algorithm will depend on what type of task:
  - Classification?
  - Clustering?
  - Regression?
  - Anomaly detection?
- We'll focus on building *classifiers.*
  - Training and validation is largely the same between types.
  - Evaluation will change.
- In practice random forests generally perform very well.





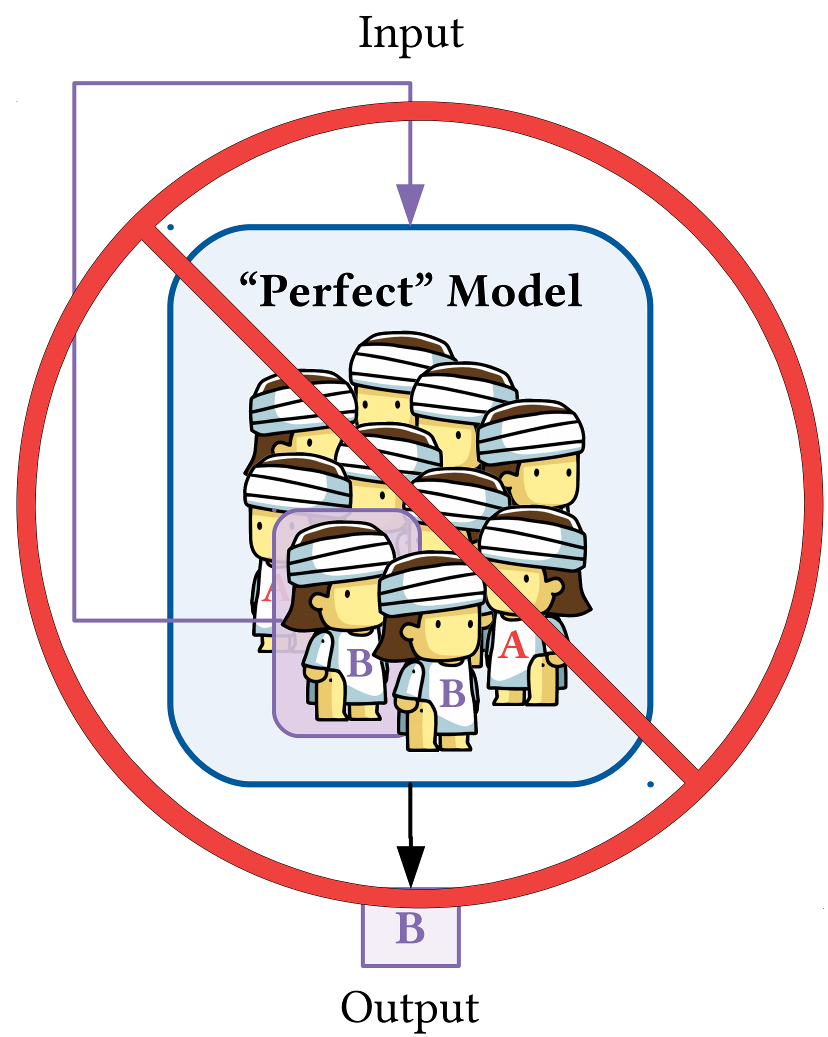
Building the "Perfect" Model

- Training on all data may fool us into training a "perfect" model: simply return the class associated with each input.
  - 100% accuracy! A+!
  - Except…



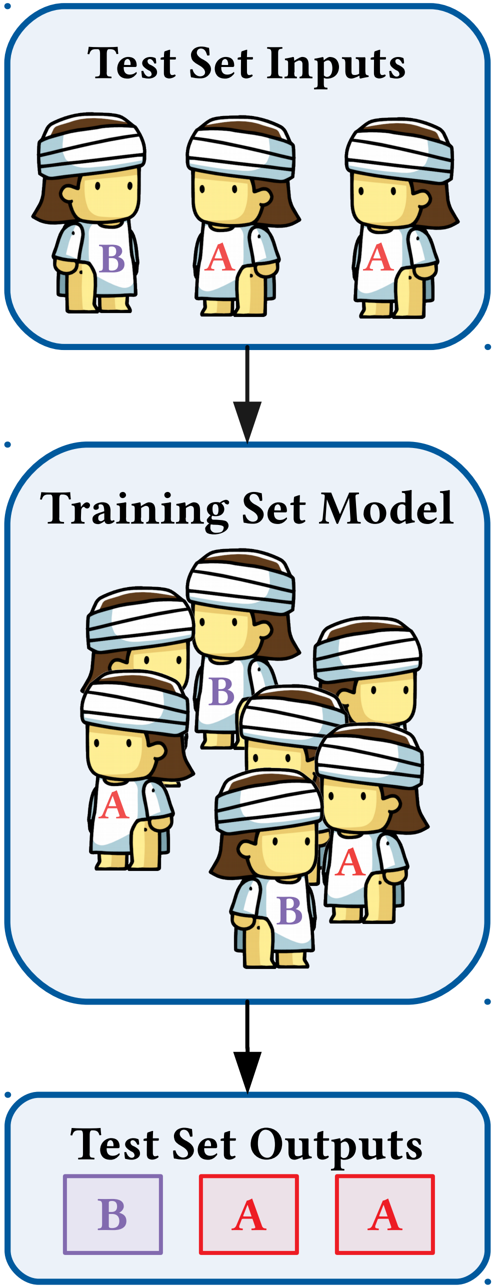
"Perfect" Models Overfit the Data

- Training on all data may fool us into training a "perfect" model: simply return the class associated with each input.
  - 100% accuracy! A+!
  - Except…
- …applying model to new data often yields poor performance due to *model overfitting*.



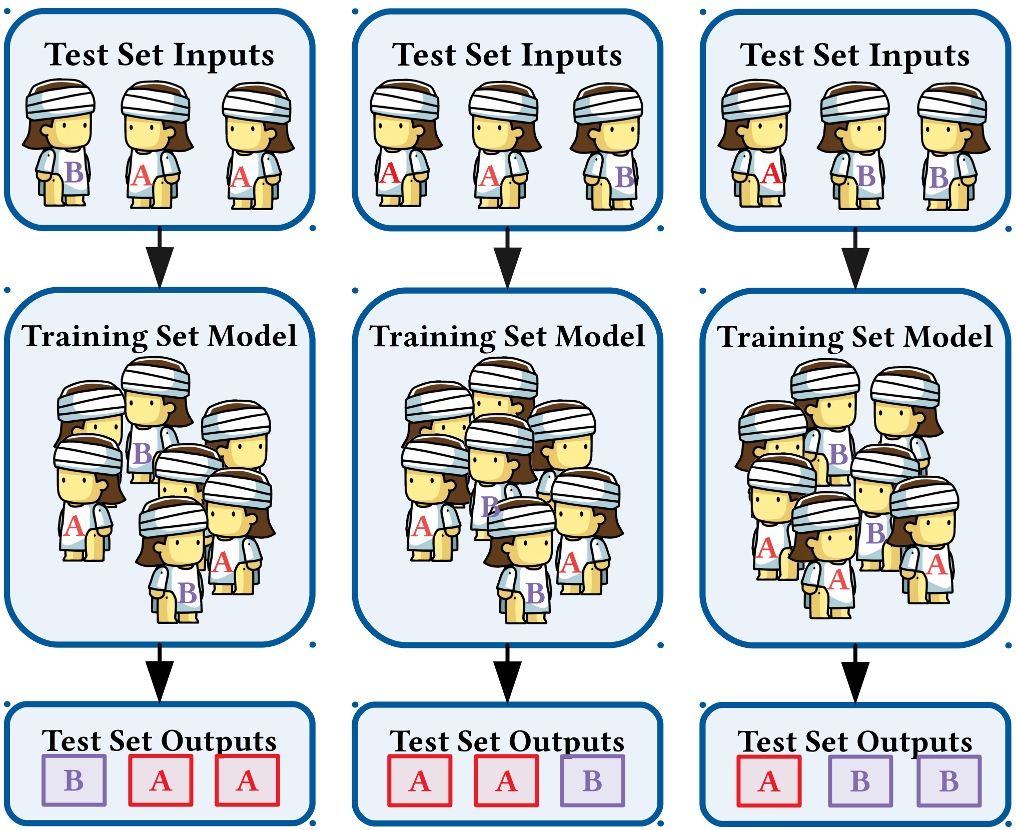
Train and Test Models on Different Datasets

- Training on all data may fool us into training a “perfect” model: simply return the class associated with each input.
  - 100% accuracy! A+!
  - Except…
- …applying model to new data often yields poor performance due to *model overfitting*.
- Instead, train on one subset and test on another to estimate expected performance on *new* data. This forces us to train models that *generalize*.



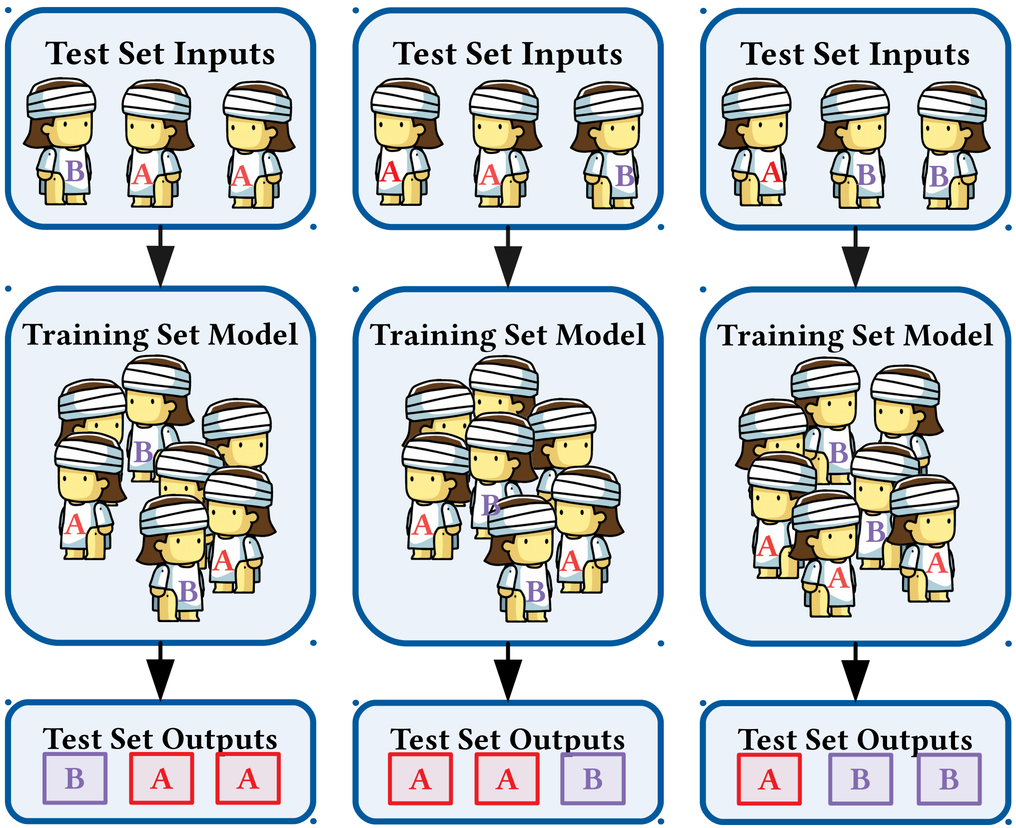
Validate on Multiple Train and Test Splits

- Training on all data may fool us into training a “perfect” model: simply return the class associated with each input.
  - 100% accuracy! A+!
  - Except…
- …applying model to new data often yields poor performance due to *model overfitting*.
- Instead, train on one subset and test on another to estimate expected performance on *new* data. This forces us to train models that *generalize*.
- Do this multiple times to determine expected performance with confidence bounds (called *cross validation*).
  - We can split by each patient in a “leave one patient out” cross validation.



Validate on Multiple Train and Test Splits

- Training on all data may fool us into training a “perfect” model: simply return the class associated with each input.
  - 100% accuracy! A+!
  - Except…
- …applying model to new data often yields poor performance due to *model overfitting*.
- Instead, train on one subset and test on another to estimate expected performance on *new* data. This forces us to train models that *generalize*.
- Do this multiple times to determine expected performance with confidence bounds (called *cross validation*).
  - We can split by each patient in a “leave one patient out” cross validation.
- This lets us compare model algorithms and instances (specific *hyper-parameter* choices).
  - We can also use cross validation to choose hyper-parameters.



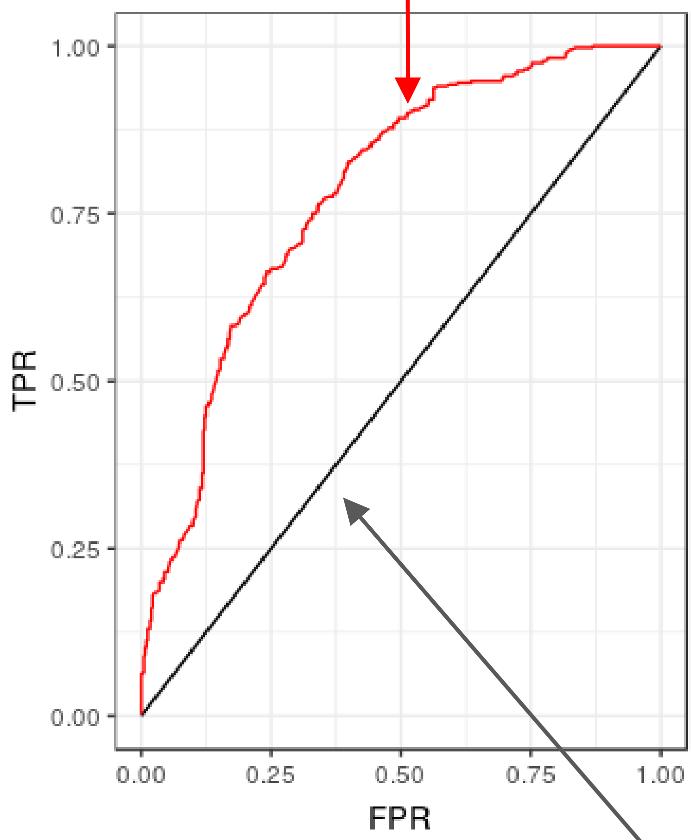
Evaluating Performance with Receiver Operating Characteristic (ROC) Curves

Introducing the ROC (Receiver Operating Characteristic) Curve

An ROC curve characterizes the performance tradeoffs made when tuning a classifier threshold.

We generally include at least a **random choice model** and one or more **other models** we want to compare.

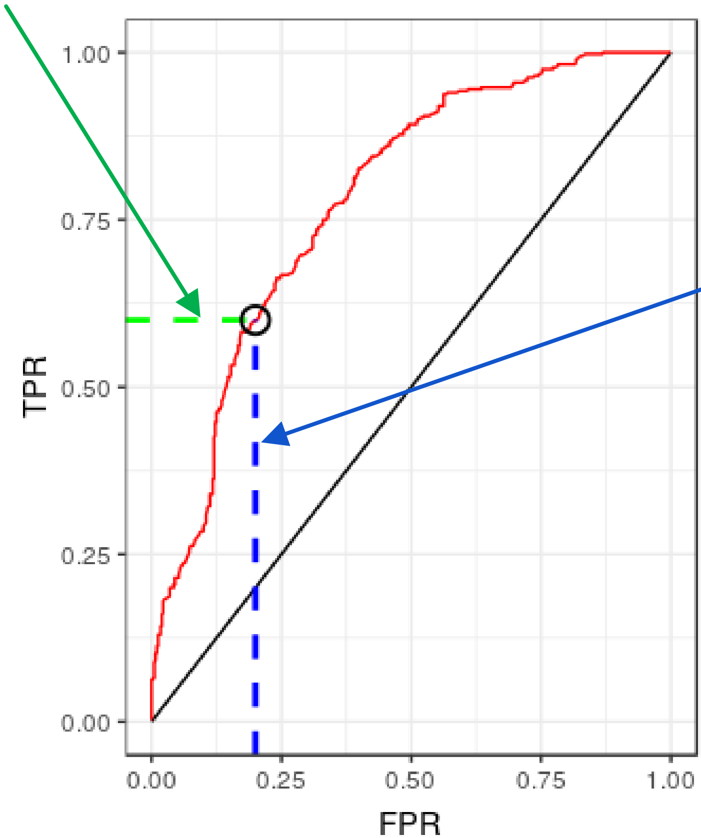
This is the model we want to compare with random guessing.



This model (call it "Random") chooses a class at random with uniform probability.

Purpose an ROC Curve

TPR: What fraction of the positive cases did we correctly identify?



FPR: How often are we incorrectly alerting of a condition that is not really present?

(Or: How much do the nurses hate the new monitor?)

An ROC curve characterizes the performance tradeoffs made when tuning a classifier threshold.

We generally include at least a **random choice model** and one or more **other models** we want to compare.

This curve characterizes the tradeoff between improving true positive rate (TPR) or false positive rate (FPR).

For a given FPR we can lookup the expected TPR.

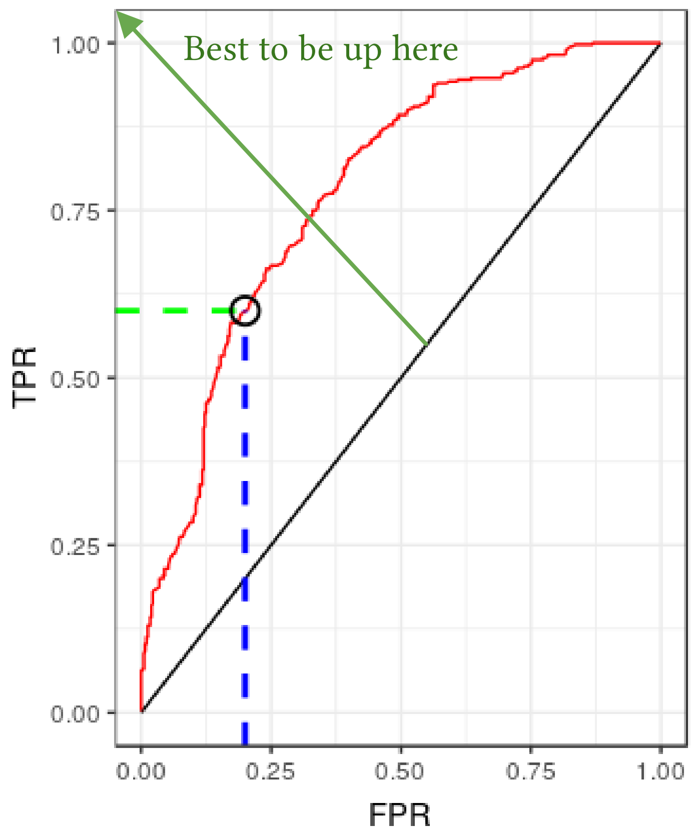
Evaluating an ROC Curve

An ROC curve characterizes the performance tradeoffs made when tuning a classifier threshold.

We generally include at least a **random choice model** and one or more **other models** we want to compare.

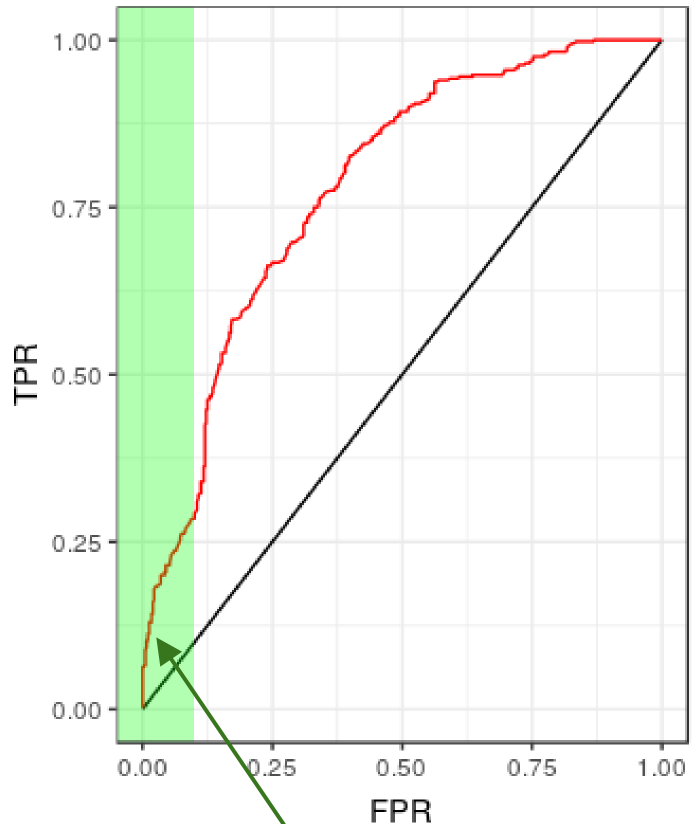
This curve characterizes the tradeoff between improving true positive rate (TPR) or false positive rate (FPR).

For a given FPR we can lookup the expected TPR.



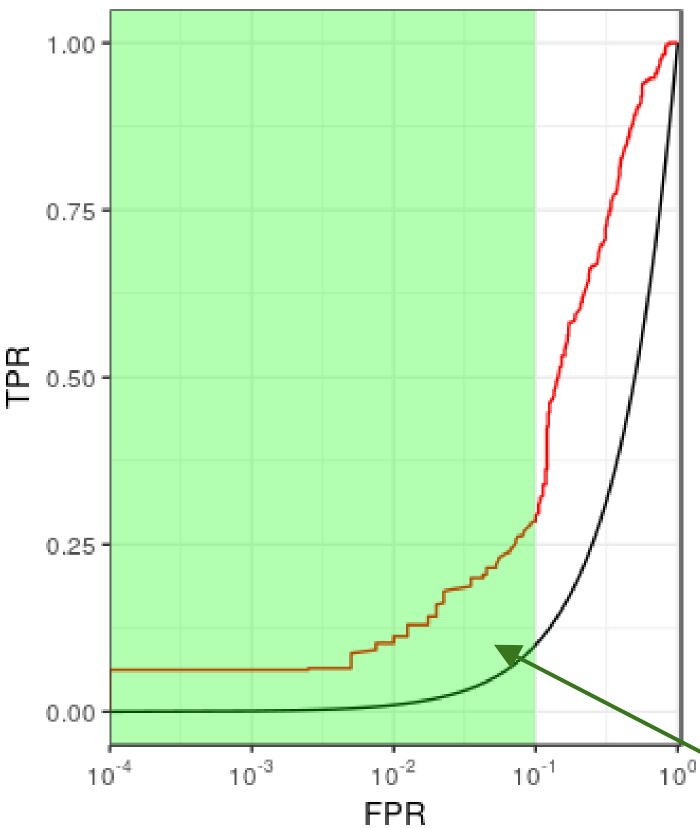
A better performing classifier will tend to move the curve toward the top left corner (i.e. more positive detections made with fewer false detections).

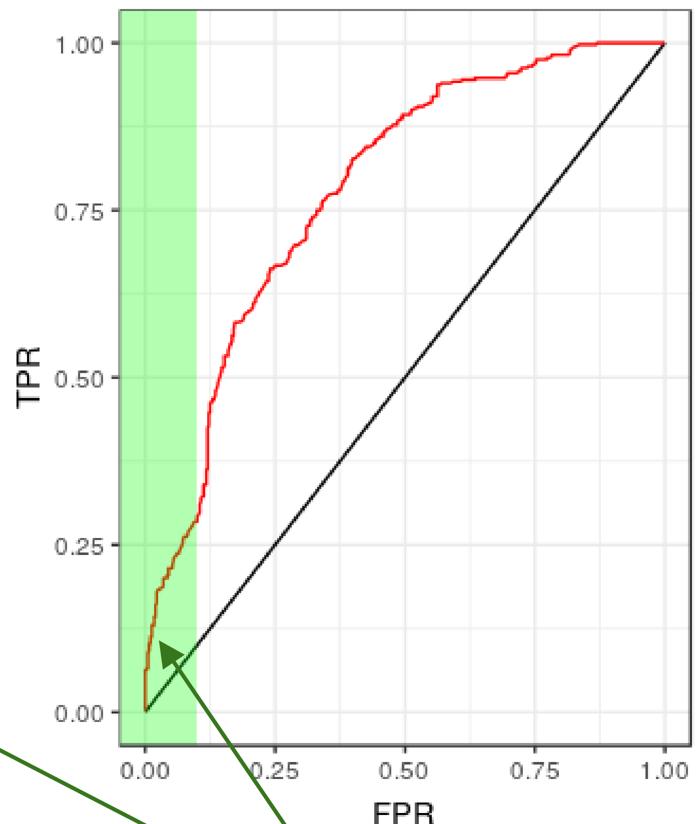
Low False Positive Rates on an ROC Curve



We are often most interested in the low FPR range in operation...

We want to look at this region.
Low FPR = Fewer false alarms.





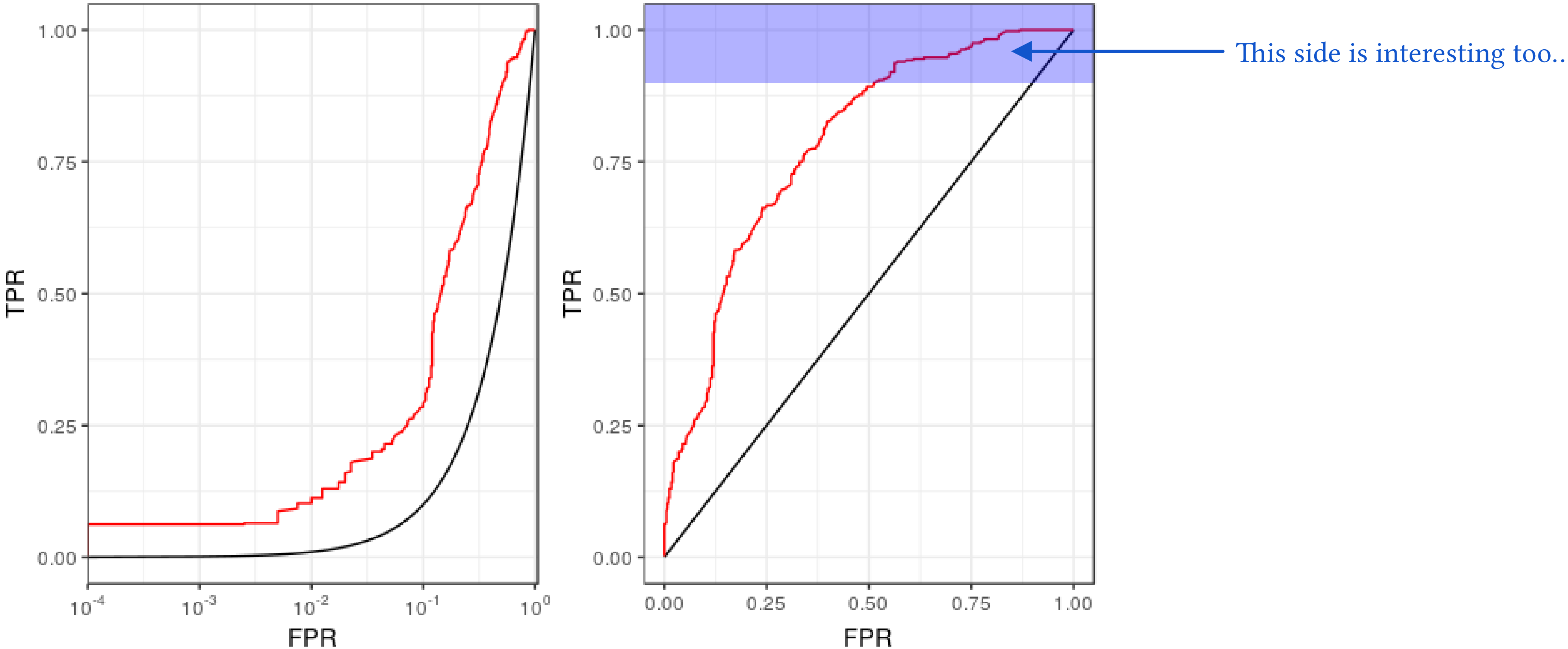
Now it's much clearer.

We are often most interested in the low FPR range in operation...

...so we plot FPR on the log scale to zoom in to the smaller values.

Low False Negative Rates on an ROC Curve

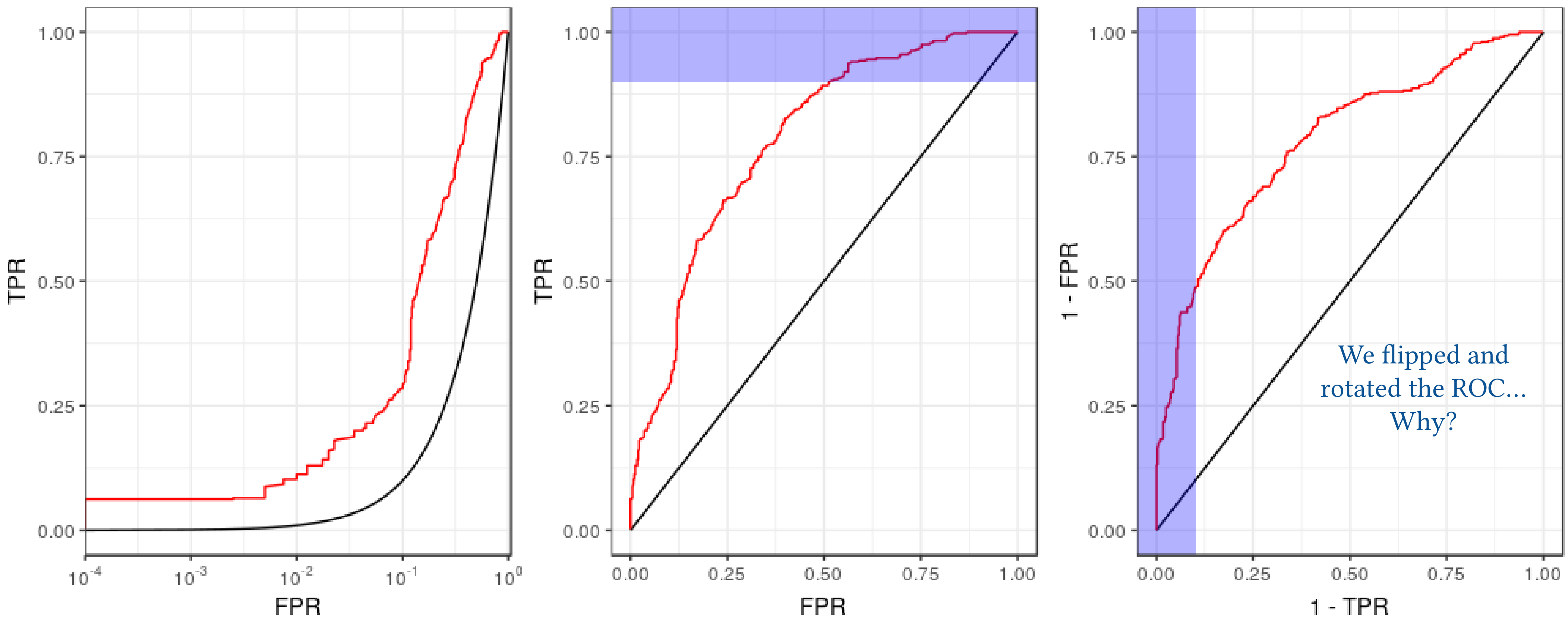
The other end of the ROC is also interesting, so we want to zoom there too...



Low False Negative Rates on an ROC Curve

The other end of the ROC is also interesting, so we want to zoom there too…

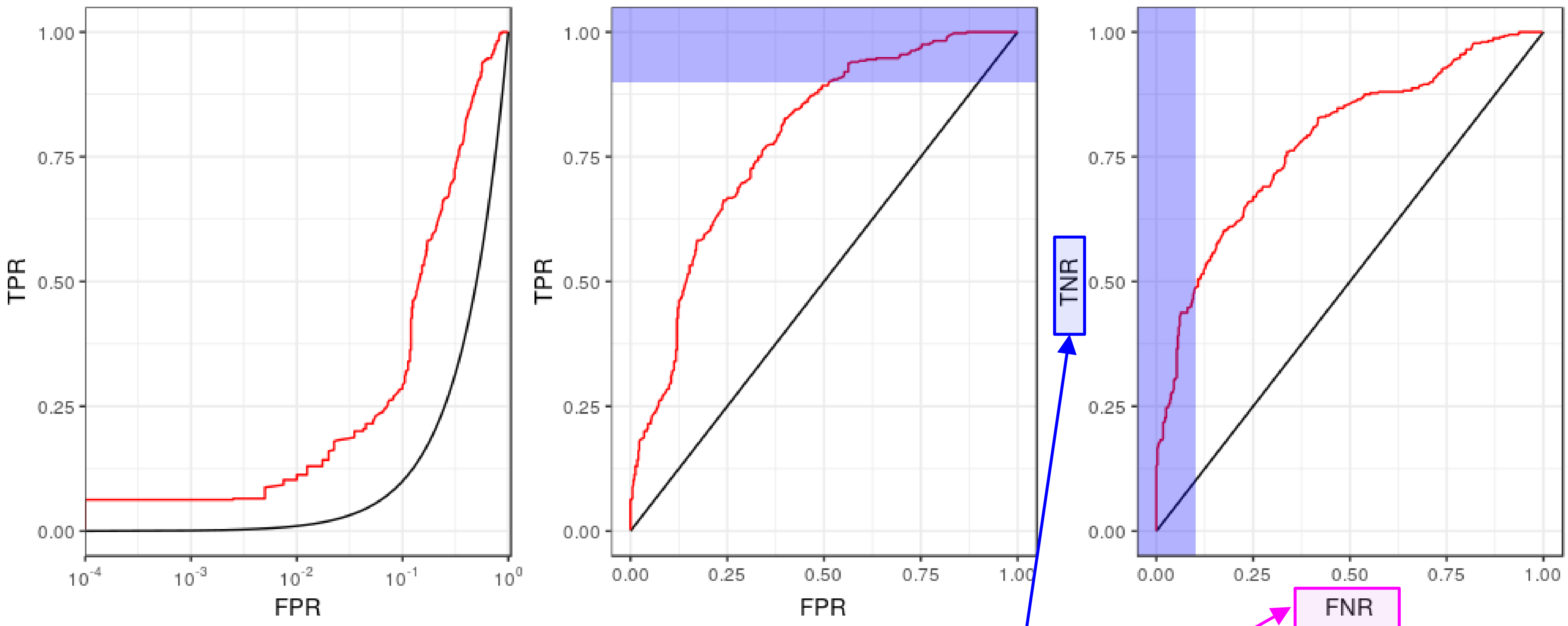
…so we can invert the rates and swap the axes…



Low False Negative Rates on an ROC Curve

The other end of the ROC is also interesting, so we want to zoom there too…

…so we can invert the rates and swap the axes…



TNR: How much time do we spend on patients who need it by avoiding too much focus on those who don't?

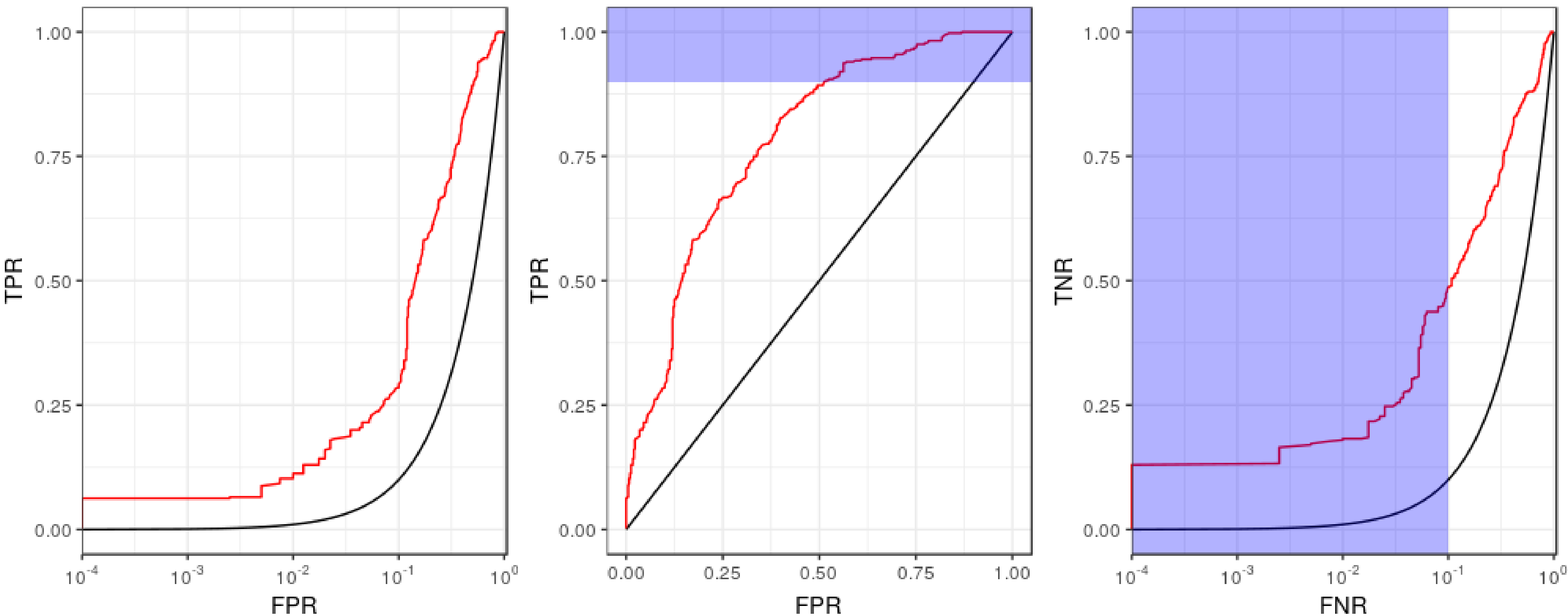
FNR: How often do we miss something potentially really bad?

Note that:

- 1-FPR=**TNR**
- 1-TPR=**FNR**

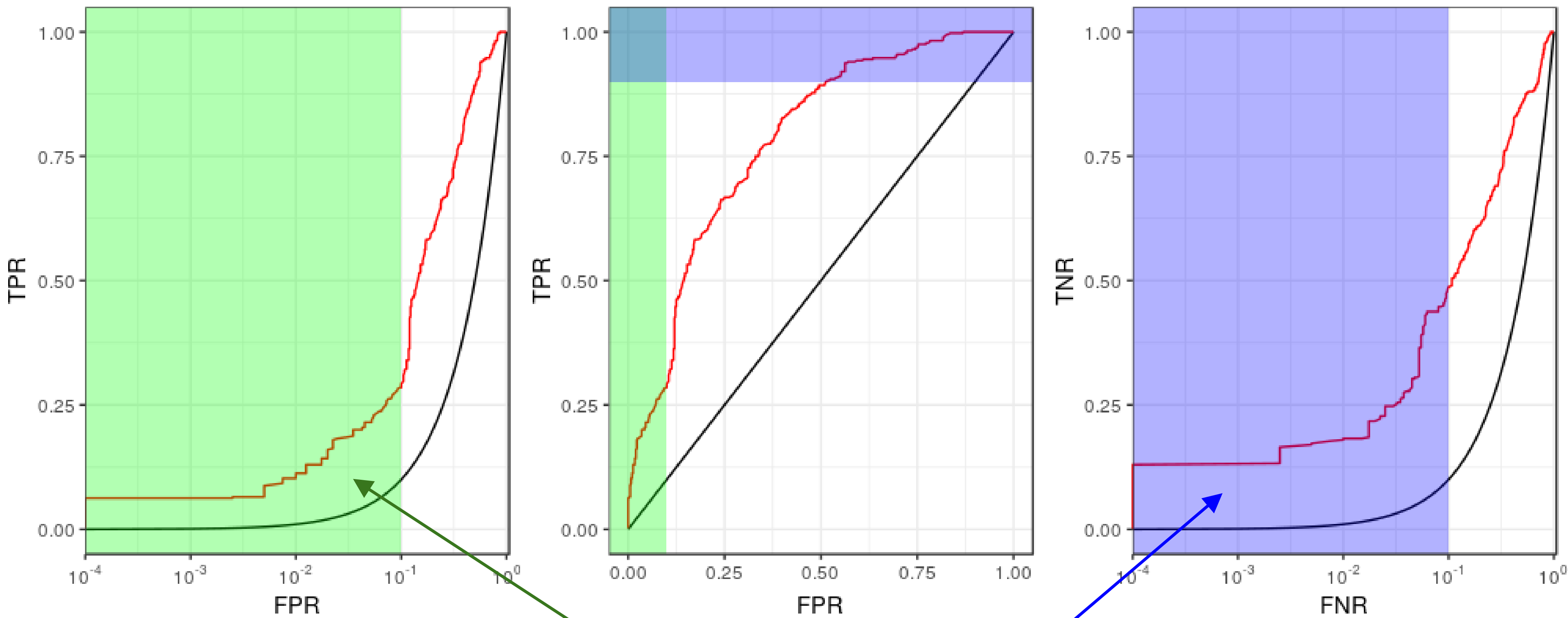
Low False Negative Rates on an ROC Curve

…then plot the false negative rate (FNR) on the log scale.



ROC Curve

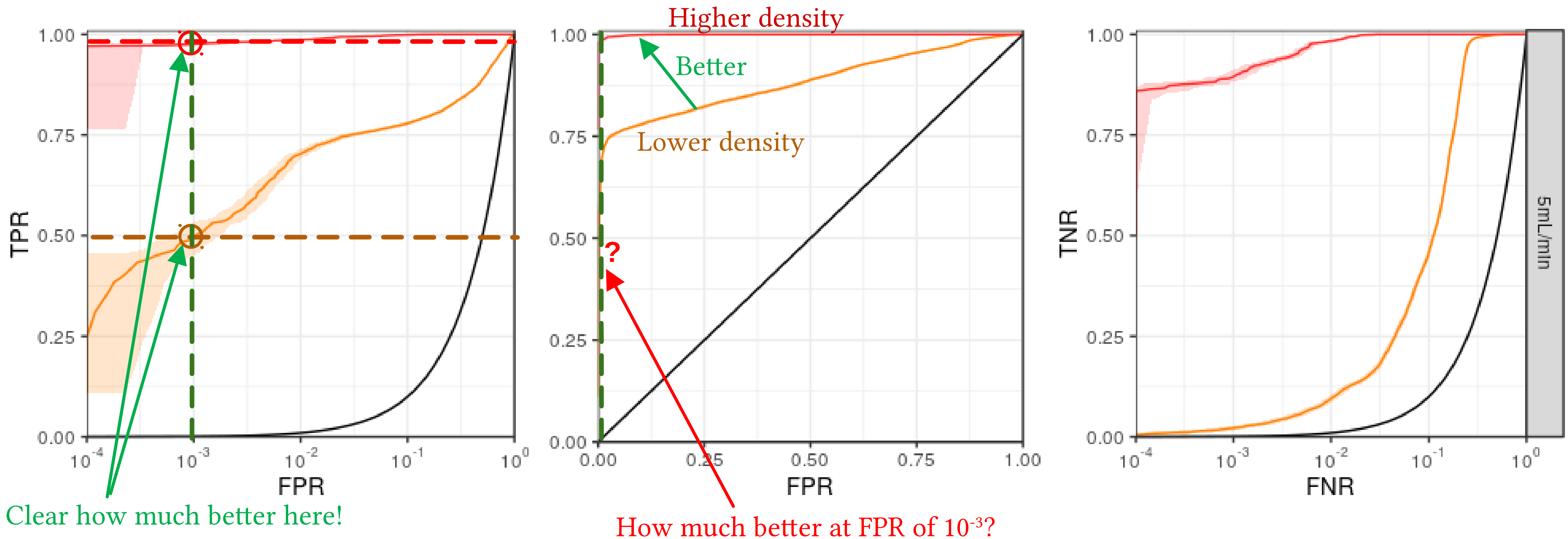
...then plot the false negative rate (FNR) on the log scale.

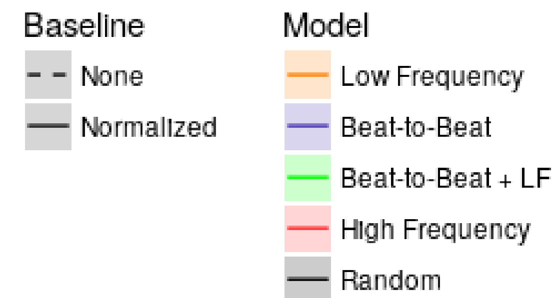


Now we see both interesting regions clearly!

Case Study: Higher Granularity in Data Improves Detection of Hemorrhage in Pig Models

ROC curves for two different hemorrhage detection models



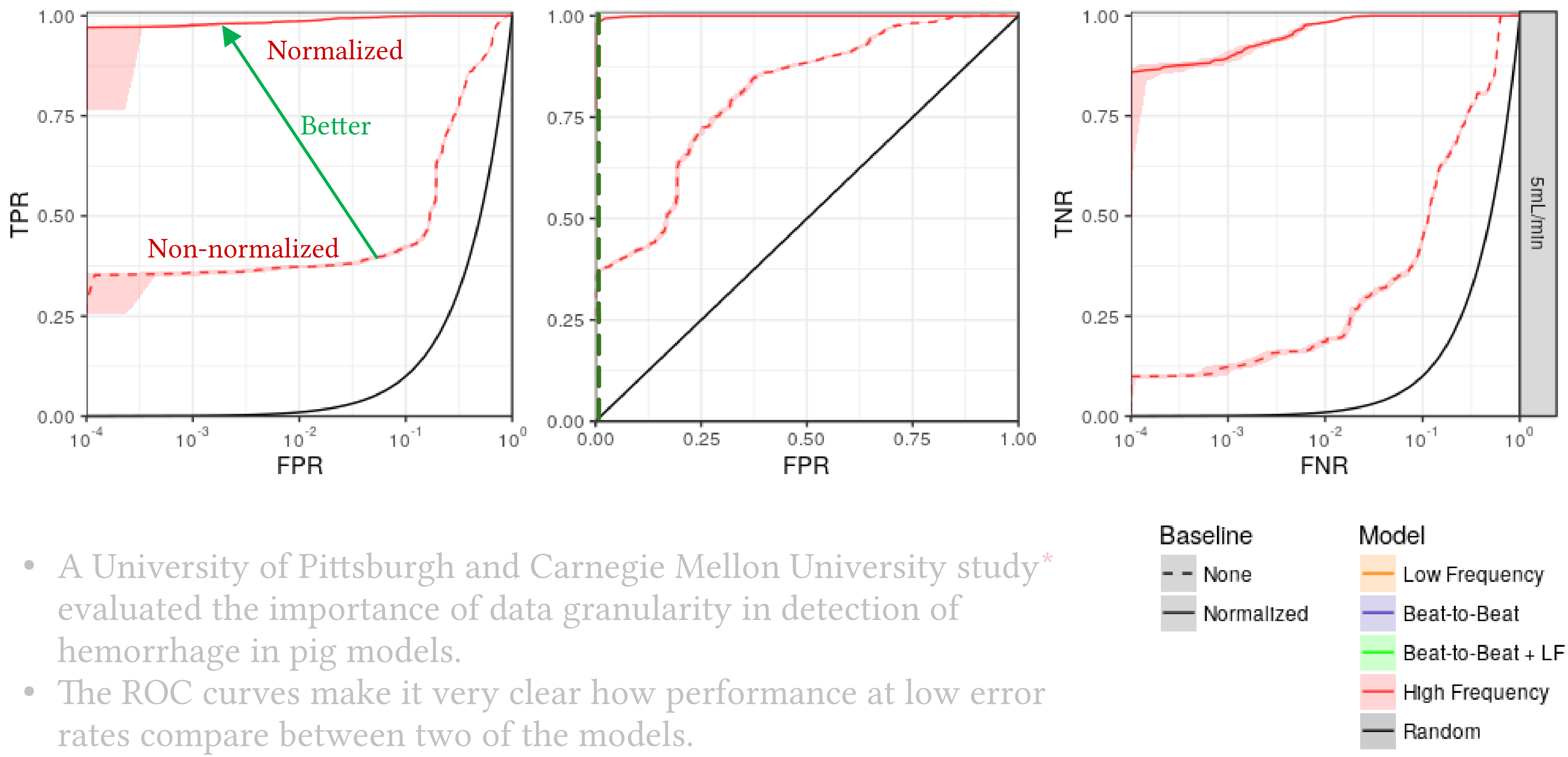


- A University of Pittsburgh and Carnegie Mellon University study* evaluated the importance of data granularity in detection of hemorrhage in pig models.
- The ROC curves make it very clear how performance at low error rates compare between two of the models.

* (In progress) Wertz et al. *Increasing sampling frequency and referencing to baseline improve hemorrhage detection.* 2018.

Case Study: Personal Baseline Normalization Improves Detection of Hemorrhage in Pig Models

ROC curves for the same model with and without normalized features



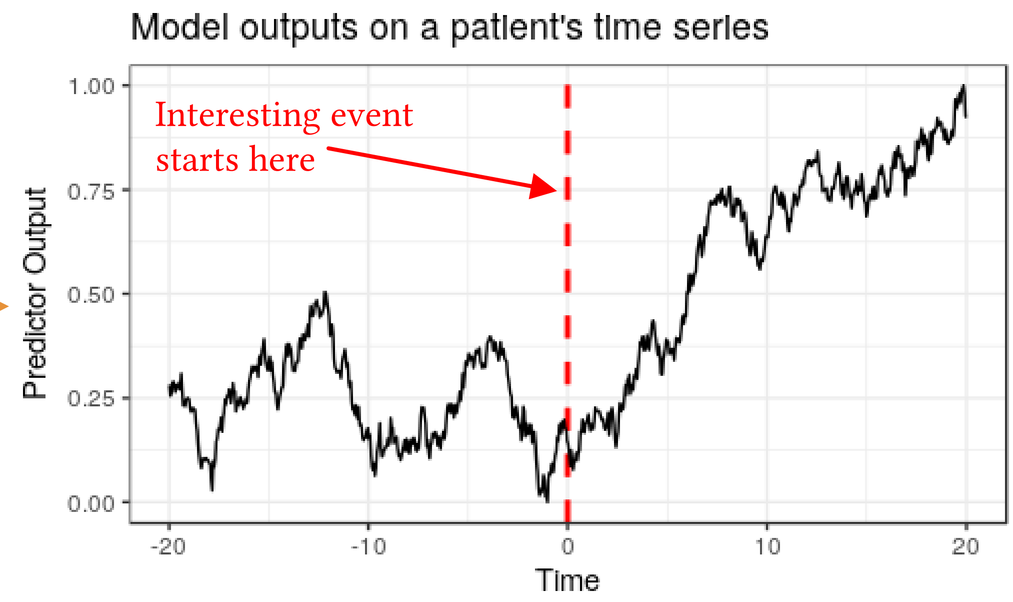
- A University of Pittsburgh and Carnegie Mellon University study* evaluated the importance of data granularity in detection of hemorrhage in pig models.
- The ROC curves make it very clear how performance at low error rates compare between two of the models.
- The study also looked at the impact of normalization on personalized baselines, showing marked improvement.

* (In progress) Wertz et al. *Increasing sampling frequency and referencing to baseline improve hemorrhage detection.* 2018.

Evaluating Performance with Activity Monitoring Operating Characteristic (AMOC) Curves

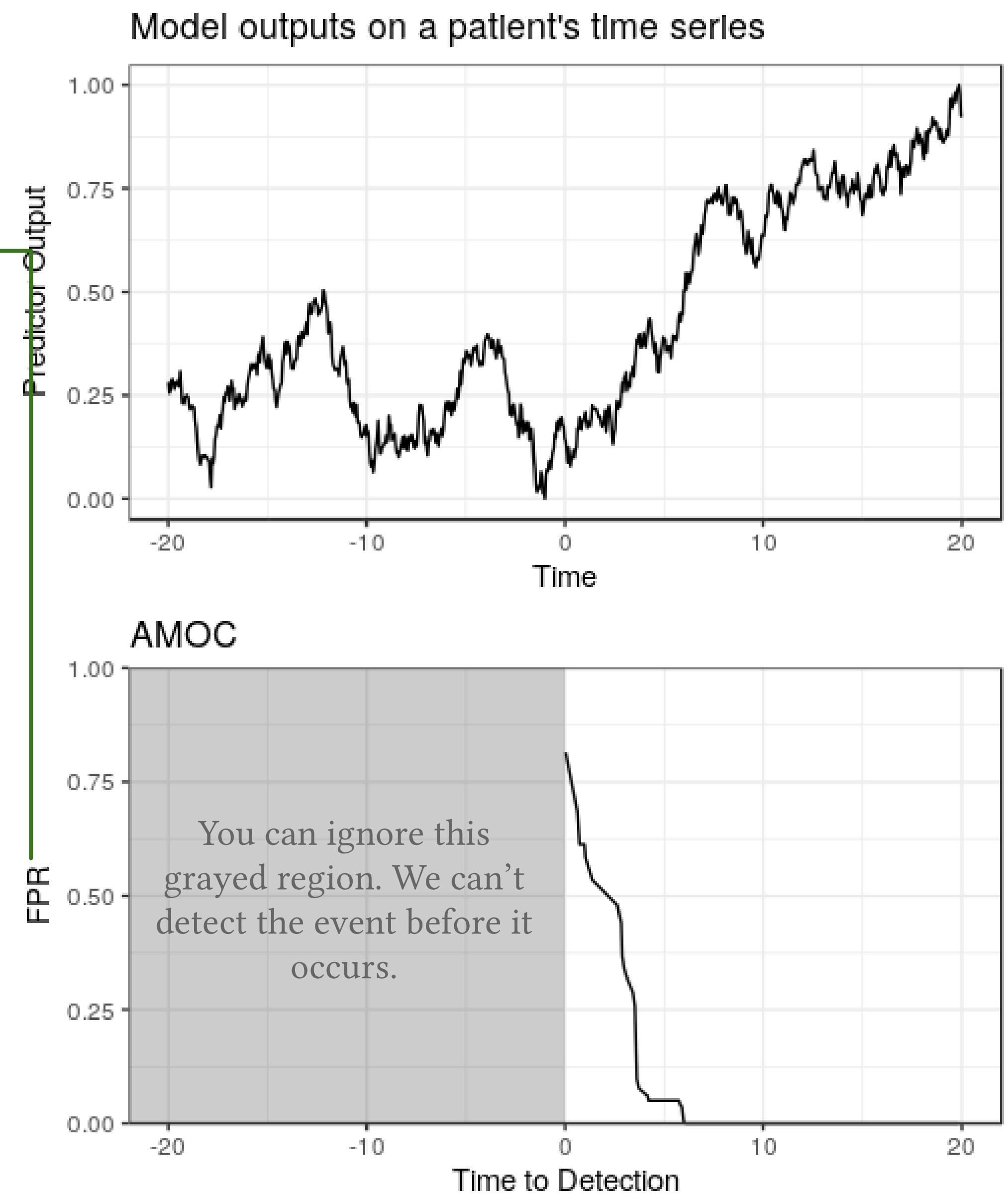
Purpose of the AMOC (Activity Monitoring Operating Characteristic) Curve

- Given a time series of predictor outputs generated by our model…



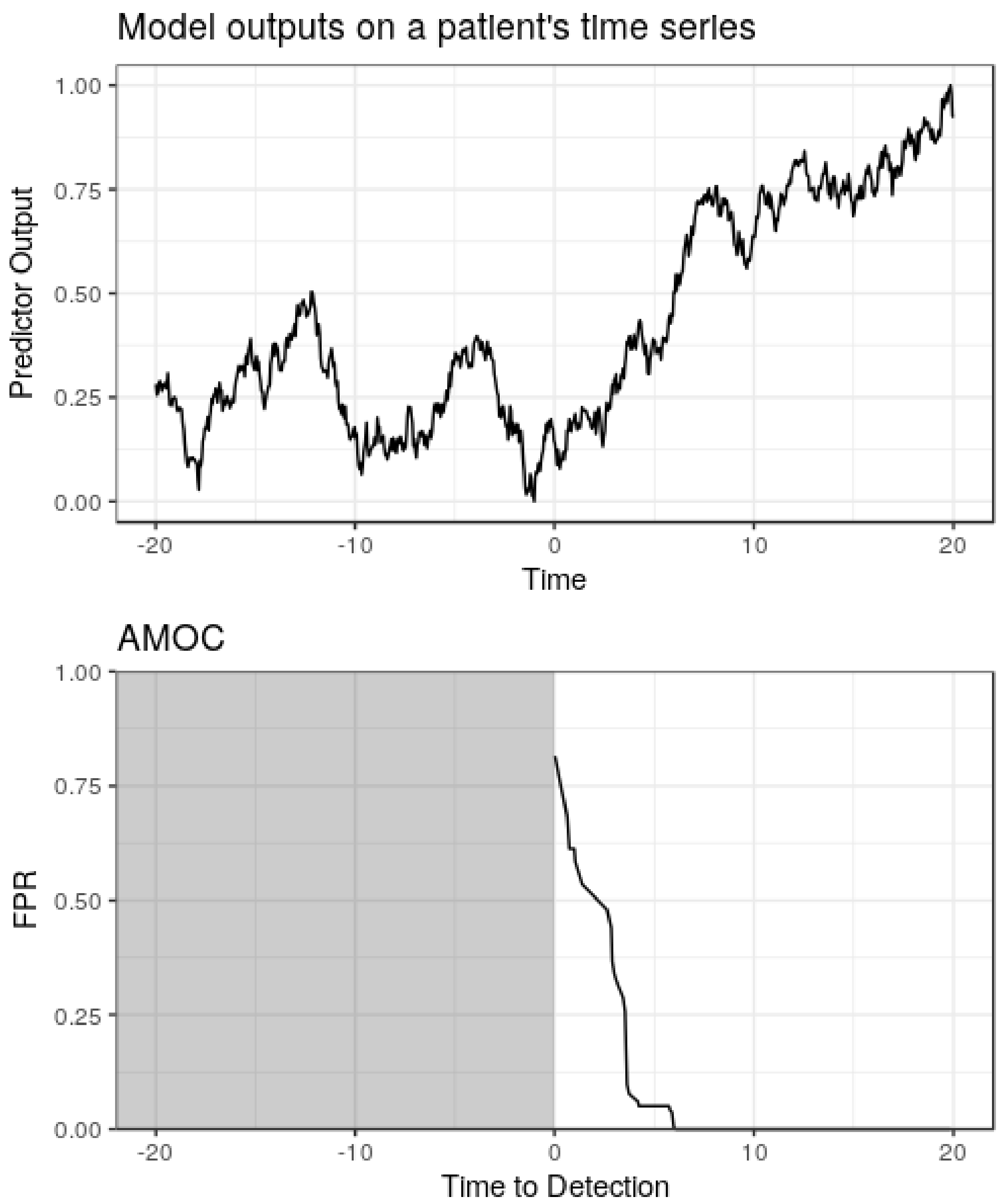
Purpose of the AMOC Curve

- Given a time series of predictor outputs generated by our model…
- …we want to characterize the **tradeoff** between detection latency (time to detection) and false alarms (FPR).



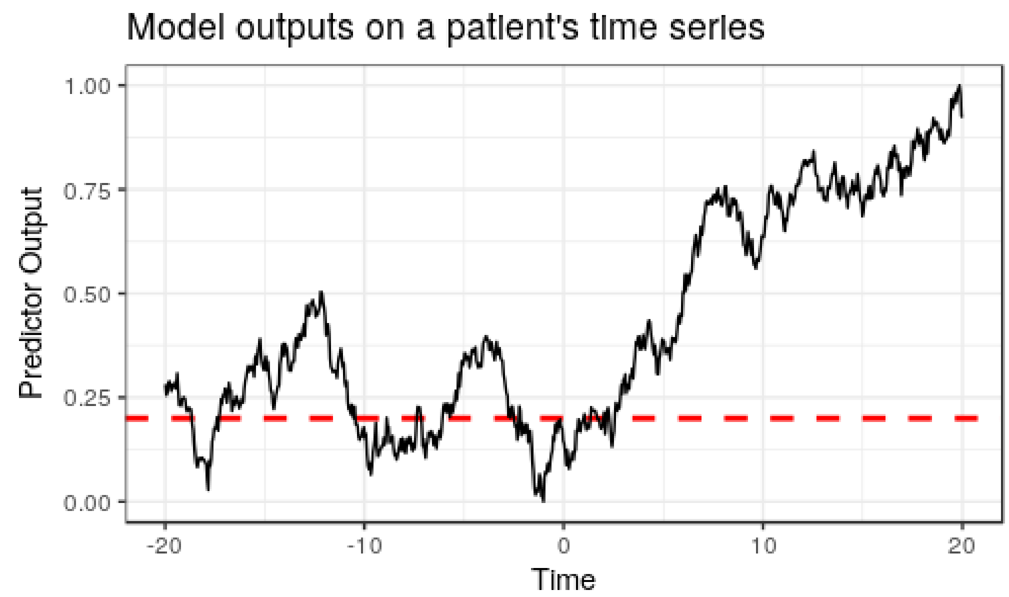
Computing an AMOC Curve

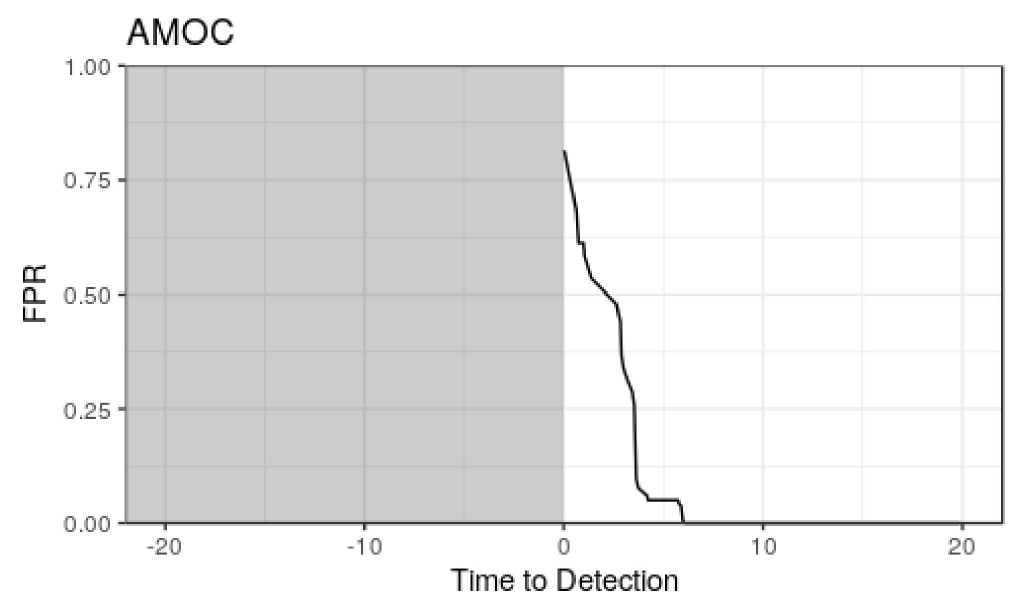
- Given a time series of predictor outputs generated by our model...
- ...we want to characterize the **tradeoff** between detection latency (time to detection) and false alarms (FPR).
- How do we compute this?



Computing an AMOC Curve

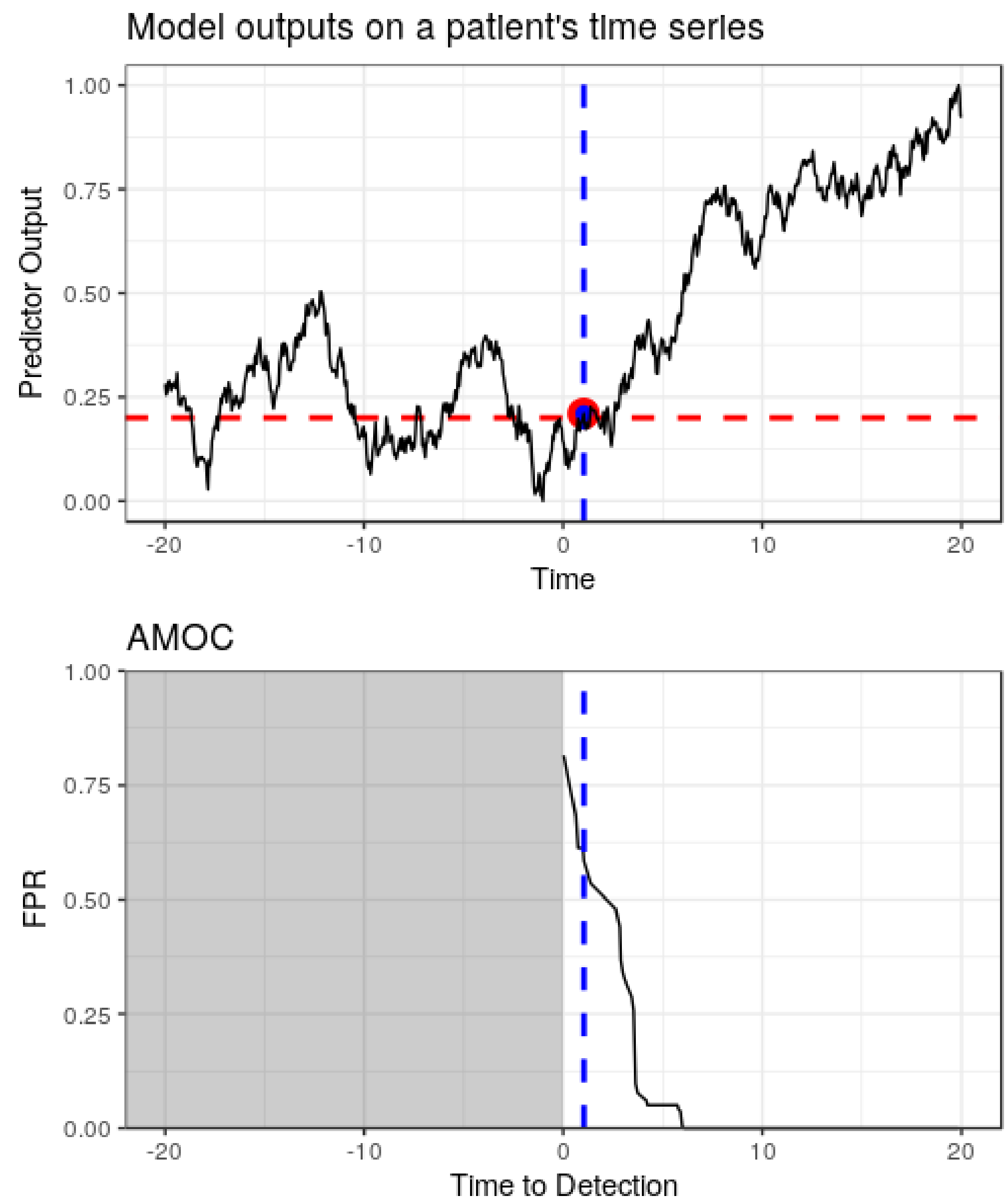
- Given a time series of predictor outputs generated by our model…
- …we want to characterize the **tradeoff** between detection latency (time to detection) and false alarms (FPR).
- How do we compute this?
  - Call a “detection” an output greater or equal to 0.2.





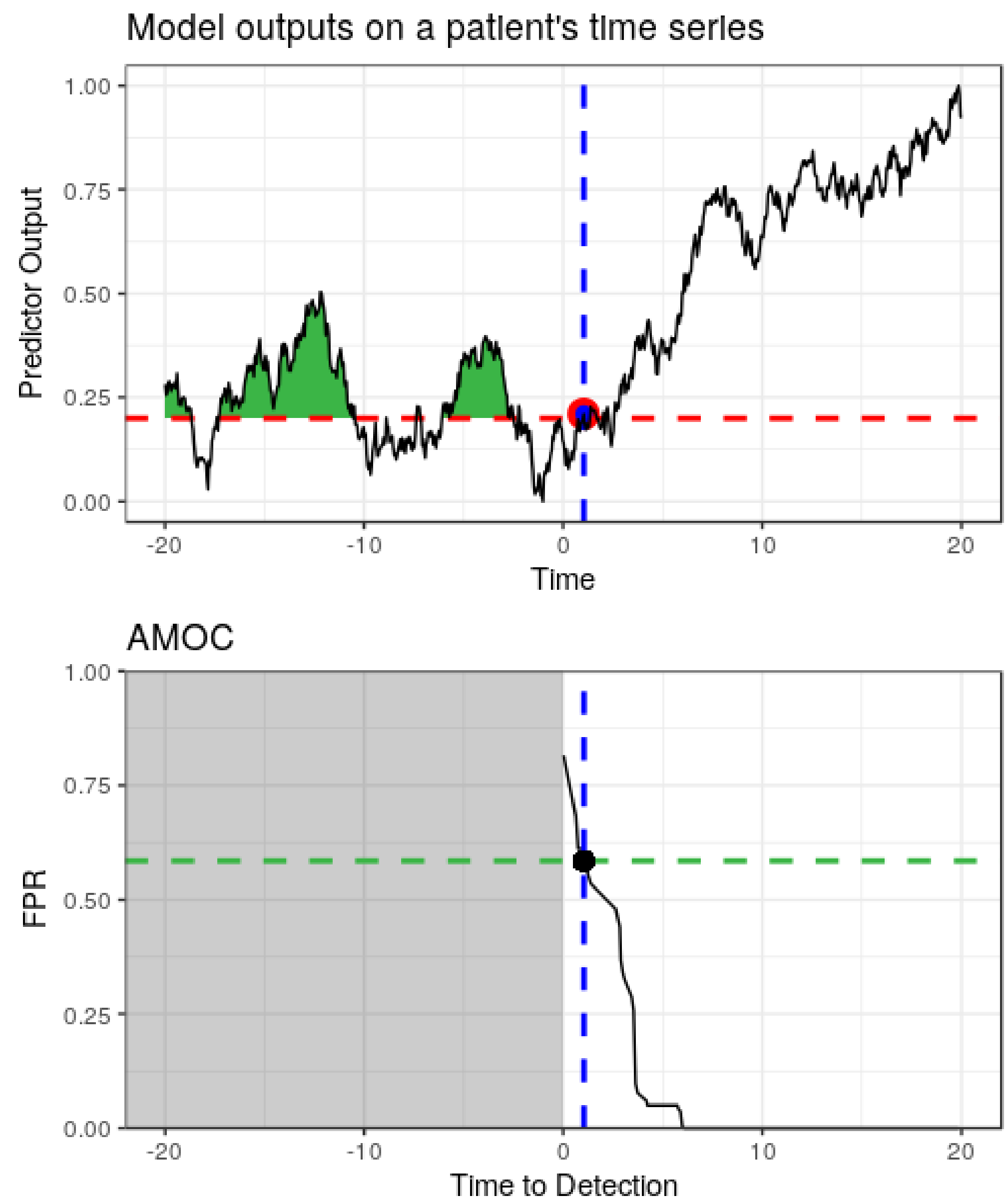
Computing an AMOC Curve

- Given a time series of predictor outputs generated by our model…
- …we want to characterize the **tradeoff** between detection latency (time to detection) and false alarms (FPR).
- How do we compute this?
  - Call a "detection" an output greater or equal to 0.2. Assigning this threshold gives us
    - A time to detection (the first true positive).



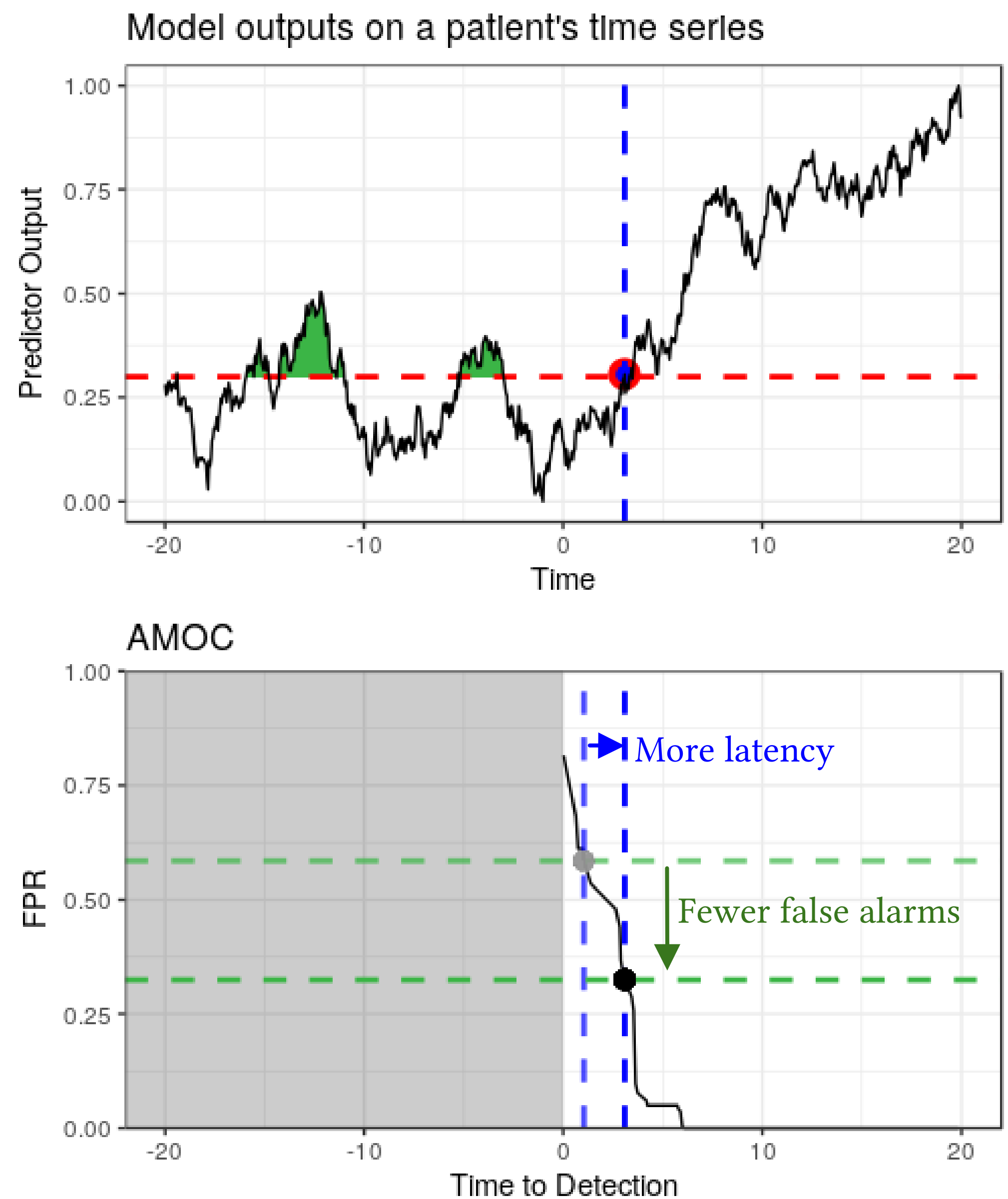
Computing an AMOC Curve

- Given a time series of predictor outputs generated by our model…
- …we want to characterize the **tradeoff** between detection latency (time to detection) and false alarms (FPR).
- How do we compute this?
  - Call a “detection” an output greater or equal to 0.2. Assigning this threshold gives us
    - A time to detection (the first true positive).
    - A number of false positives (thus, FPR).



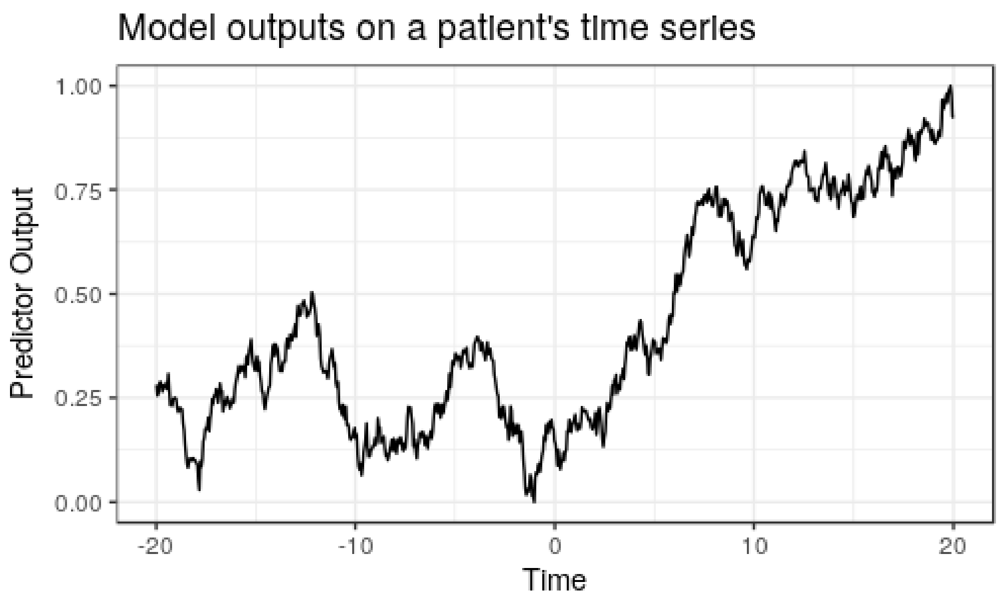
Computing an AMOC Curve

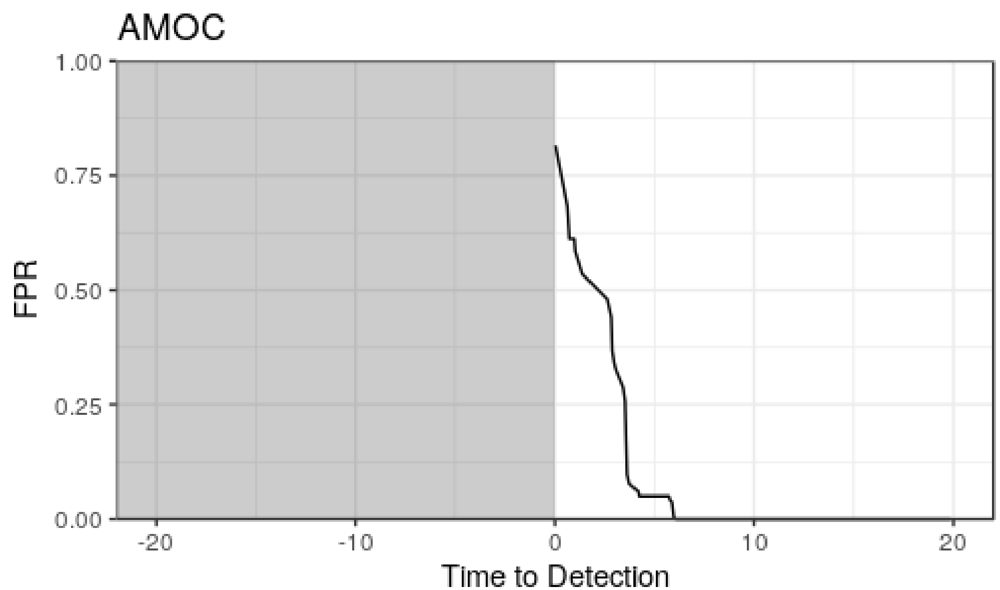
- Given a time series of predictor outputs generated by our model…
- …we want to characterize the **tradeoff** between detection latency (time to detection) and false alarms (FPR).
- How do we compute this?
  - Call a “detection” an output greater or equal to 0.2. Assigning this threshold gives us
    - A time to detection (the first true positive).
    - A number of false positives (thus, FPR).
  - Do this again for another threshold, 0.3, and now there are two points on the AMOC.



Computing an AMOC Curve

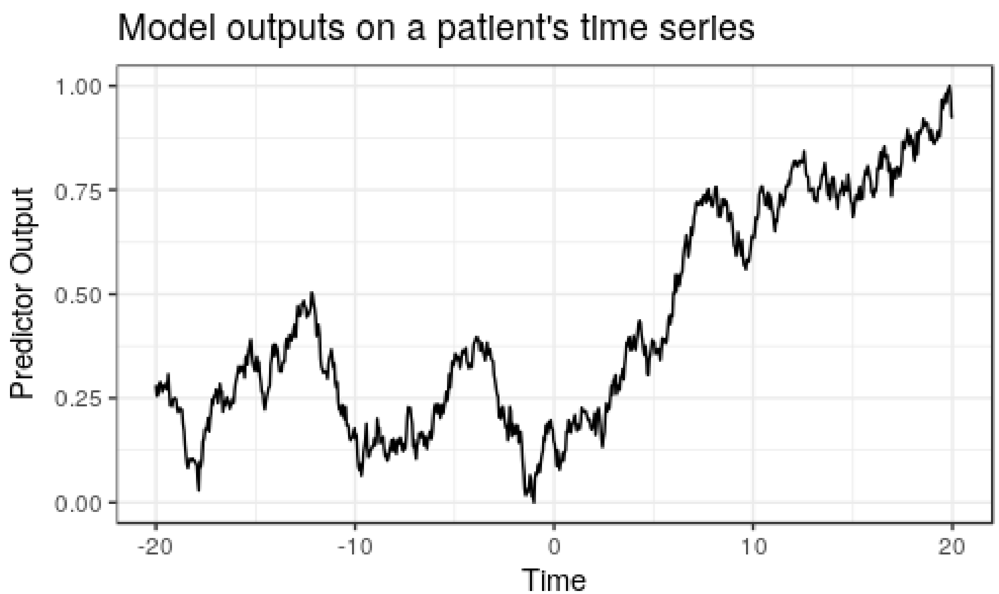
- Given a time series of predictor outputs generated by our model...
- ...we want to characterize the **tradeoff** between detection latency (time to detection) and false alarms (FPR).
- How do we compute this?
  - Call a "detection" an output greater or equal to 0.2. Assigning this threshold gives us
    - A time to detection (the first true positive).
    - A number of false positives (thus, FPR).
  - Do this again for another threshold, 0.3, and now there are two points on the AMOC.
  - Keep doing this for all thresholds for the complete curve.

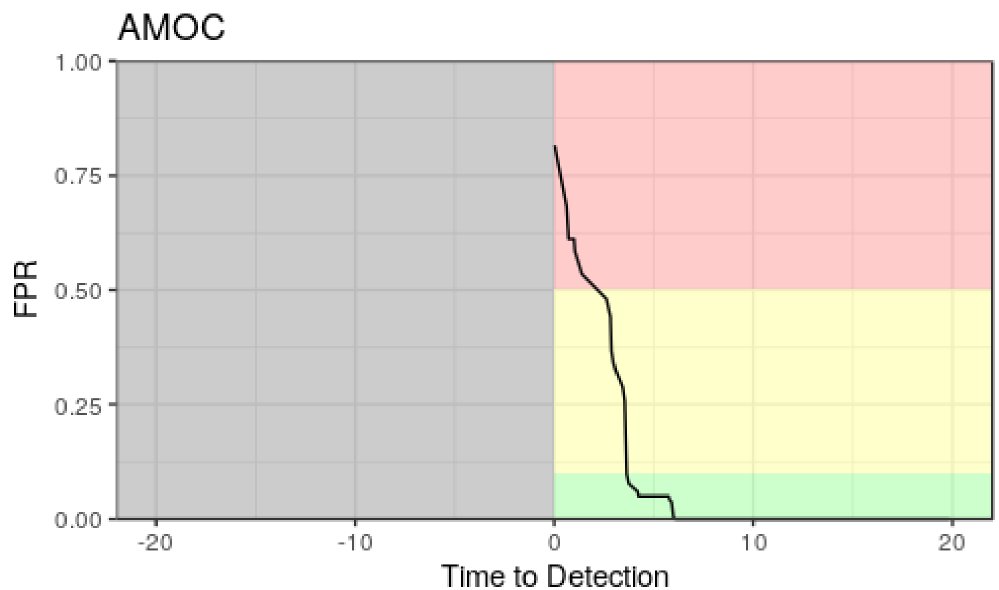




Low False Positive Rates on an AMOC Curve

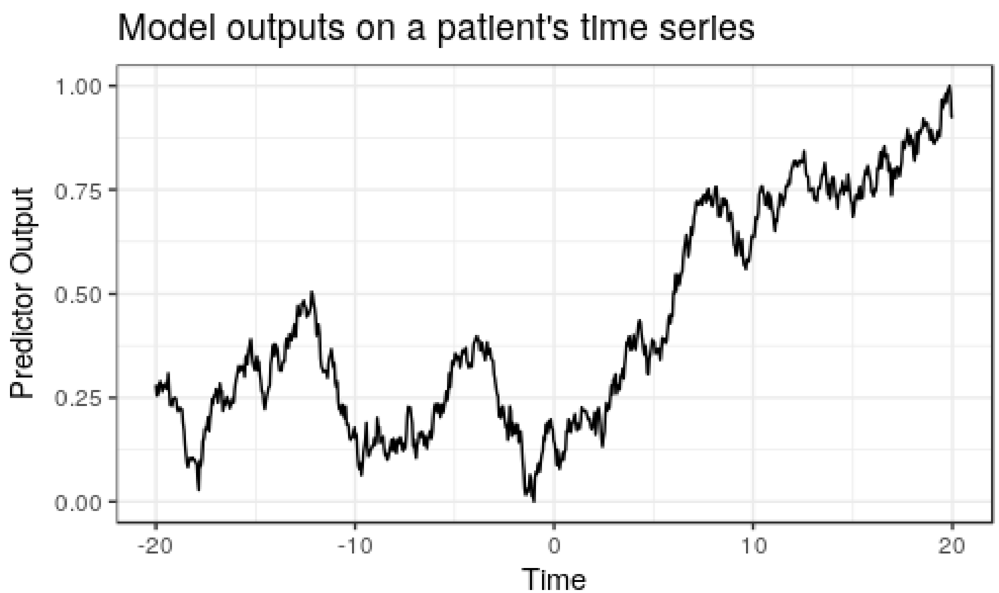
- Given a time series of predictor outputs generated by our model...
- ...we want to characterize the **tradeoff** between detection latency (time to detection) and false alarms (FPR).
- How do we compute this?
  - Call a "detection" an output greater or equal to 0.2. Assigning this threshold gives us
    - A time to detection (the first true positive).
    - A number of false positives (thus, FPR).
  - Do this again for another threshold, 0.3, and now there are two points on the AMOC.
  - Keep doing this for all thresholds for the complete curve.
- Lower FPR values are generally more operationally useful...

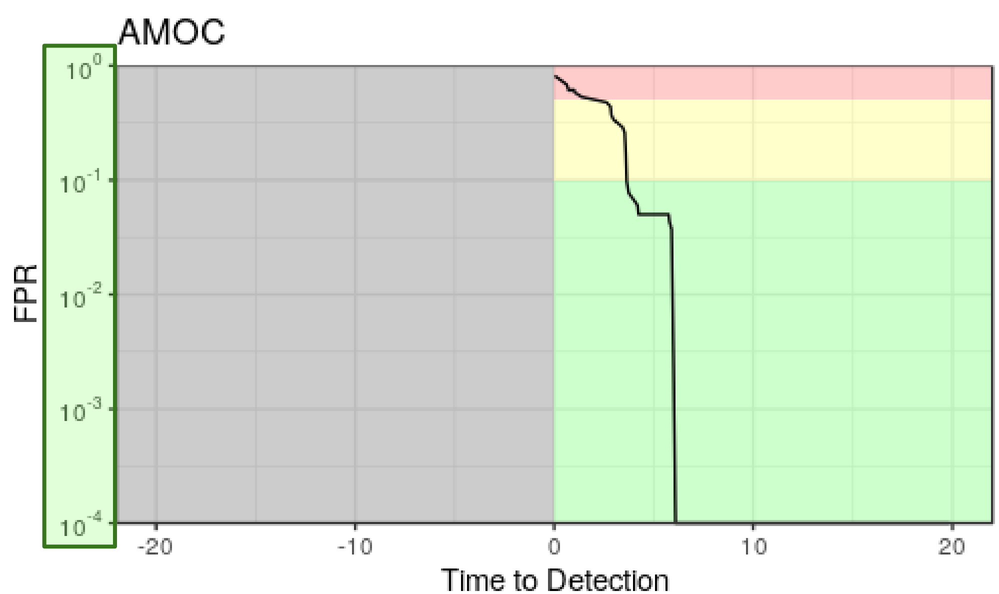




Low False Positive Rates on an AMOC Curve

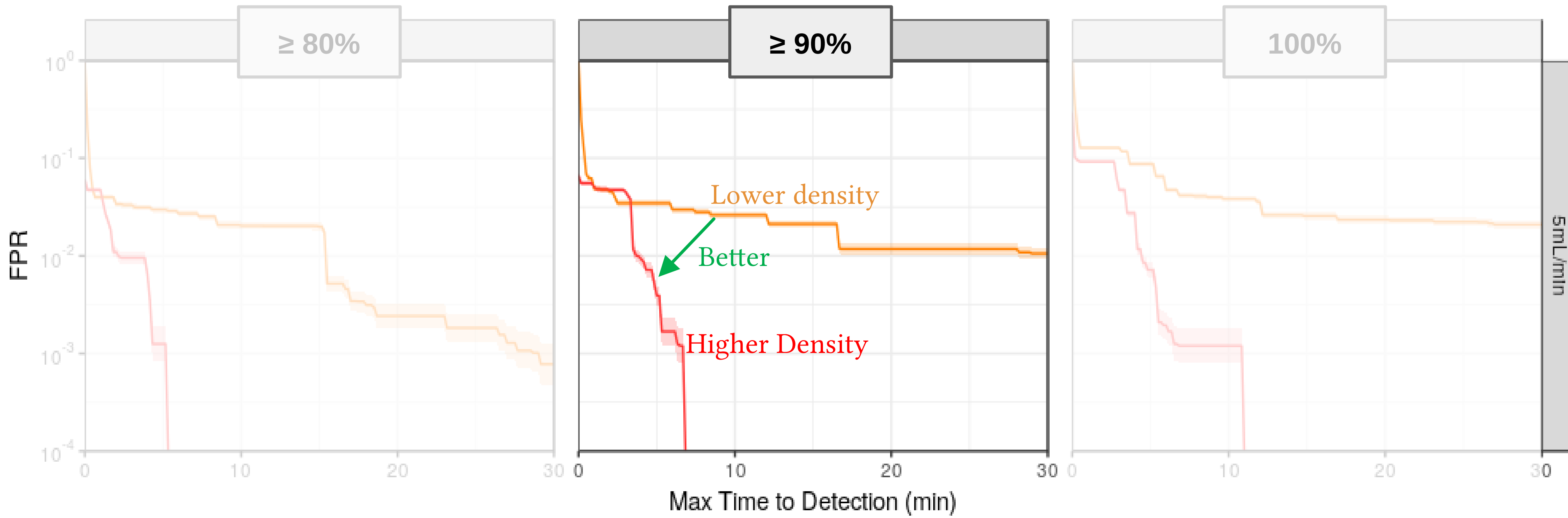
- Given a time series of predictor outputs generated by our model…
- …we want to characterize the **tradeoff** between detection latency (time to detection) and false alarms (FPR).
- How do we compute this?
  - Call a "detection" an output greater or equal to 0.2. Assigning this threshold gives us
    - A time to detection (the first true positive).
    - A number of false positives (thus, FPR).
  - Do this again for another threshold, 0.3, and now there are two points on the AMOC.
  - Keep doing this for all thresholds for the complete curve.
- Lower FPR values are generally more operationally useful… **so we put FPR on the log scale to zoom in to this region.**



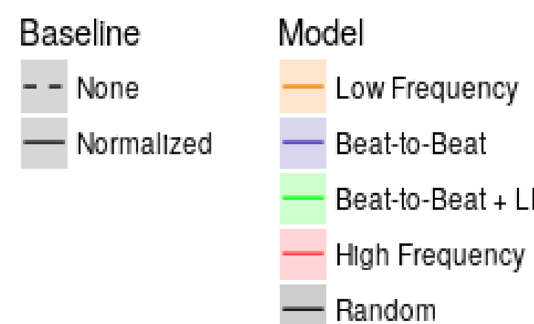


Case Study: Higher Granularity in Data Reduces Detection Latency

AMOC curves for two different hemorrhage detection models



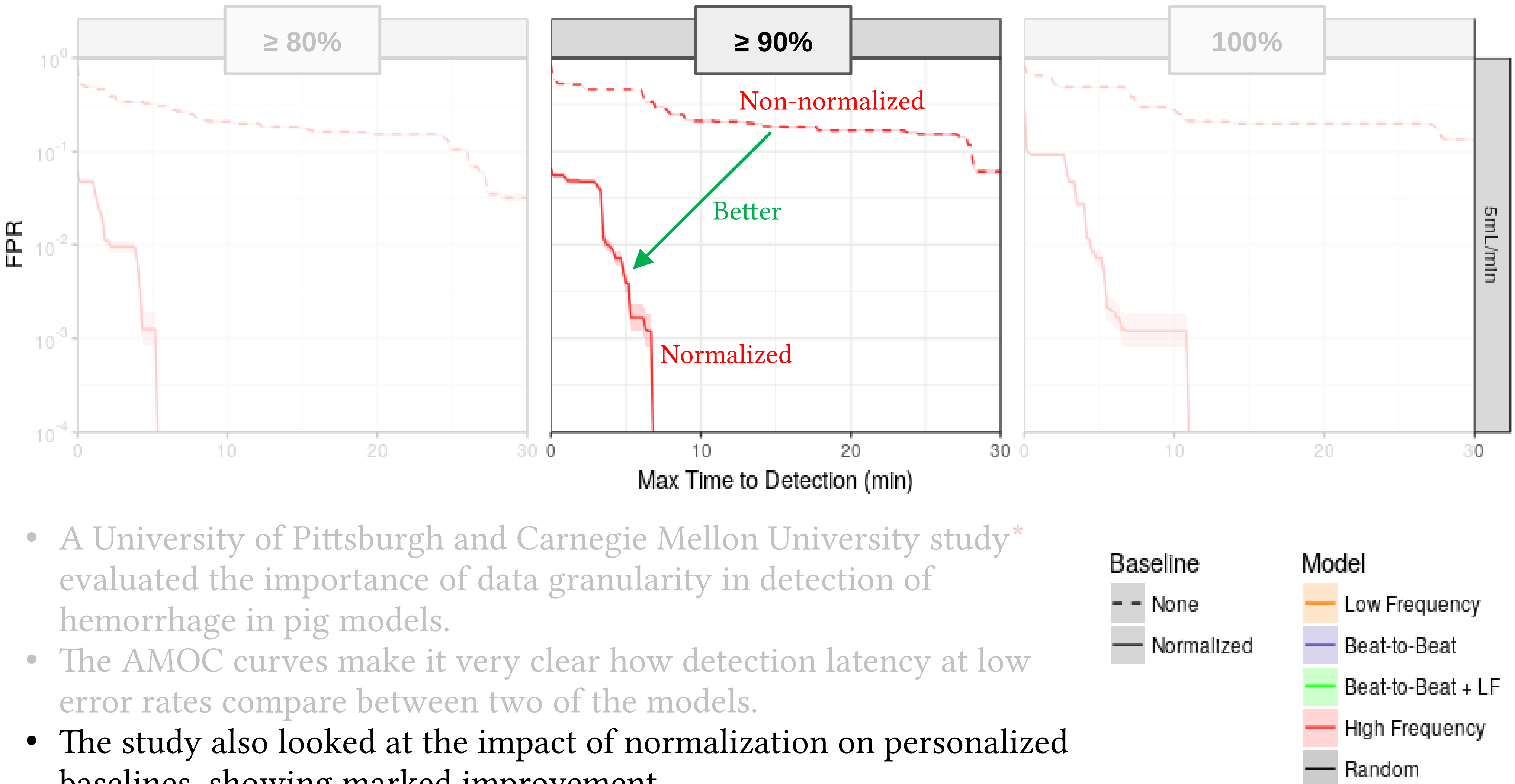
- A University of Pittsburgh and Carnegie Mellon University study* evaluated the importance of data granularity in detection of hemorrhage in pig models.
- The AMOC curves make it very clear how detection latency at low error rates compare between two of the models.



* (In progress) Wertz et al. *Increasing sampling frequency and referencing to baseline improve hemorrhage detection*. 2018.

Case Study: Personal Baseline Normalization Reduces Detection Latency

AMOC curves for the same model with and without normalized features



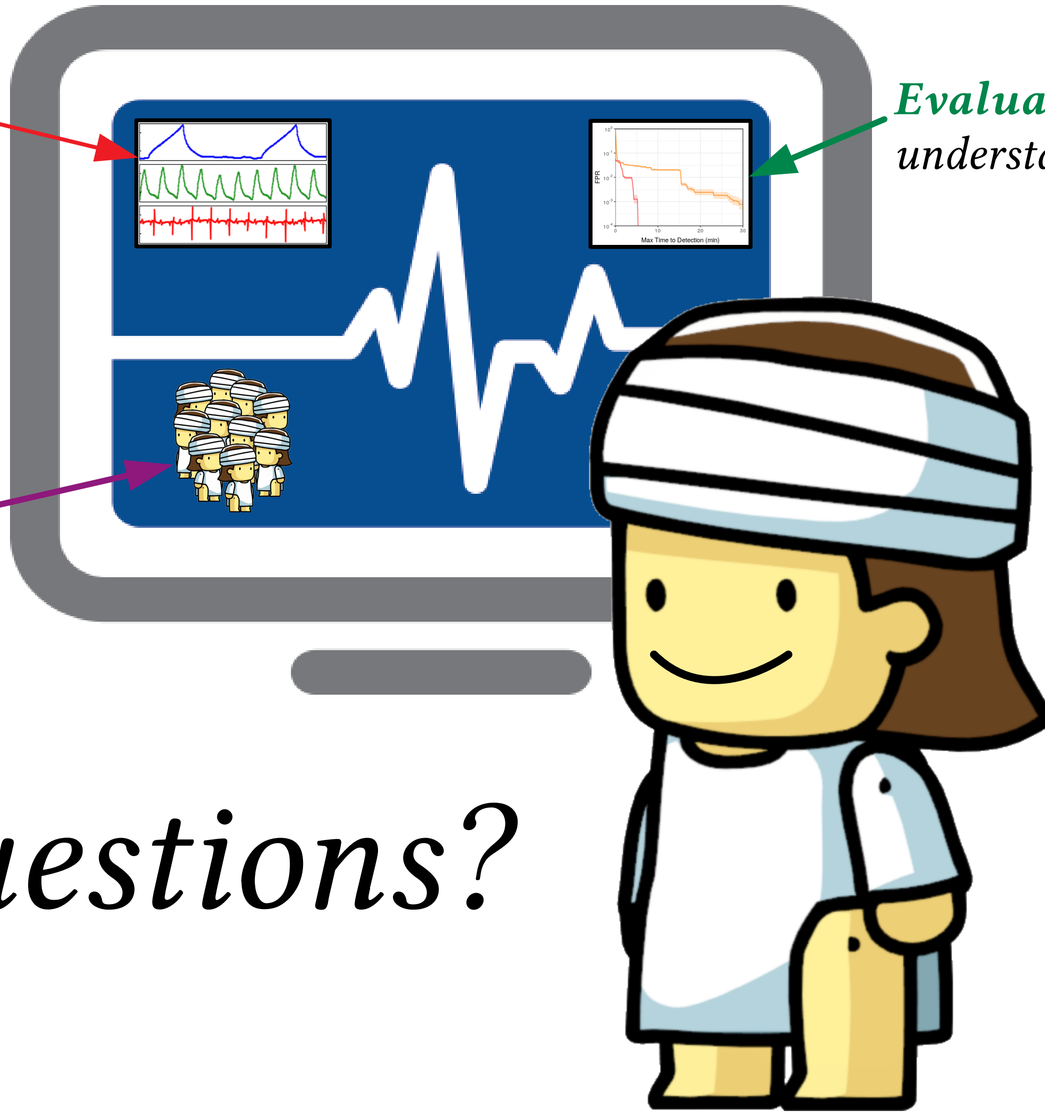
- A University of Pittsburgh and Carnegie Mellon University study* evaluated the importance of data granularity in detection of hemorrhage in pig models.
- The AMOC curves make it very clear how detection latency at low error rates compare between two of the models.
- The study also looked at the impact of normalization on personalized baselines, showing marked improvement.

* (In progress) Wertz et al. *Increasing sampling frequency and referencing to baseline improve hemorrhage detection*. 2018.

Three Steps to Building Great Models

Featurize *signals to uncover interesting information.*

Evaluate *models to understand performance.*



Validate *models through cross validation.*

Questions?