04/13 update

- Preliminary results on a small, **balanced** subsample (5000 cases, 5000 controls).
- Training set of 9000 IDs, test set of 1000 IDs.
- Imputed missing variables with median.
- Lab data (blood tests, FOBT) and colonoscopies converted to longitudinal summaries (mean,max,min,...).
- Lab data restricted to time interval from 'indexdate - 3 years' to 'indexdate - 1 year'.
- No ICD code information, no medication information, no smoking status records (smoking status at index is in demographic variables).
- Logistic regression.

Predictors (p = # of predictors)	**Training AUC**	**Test AUC**
Demographic ($p = 10$)	0.814	0.818
Charlson ($p = 18$)	0.705	0.693
Demographic + Charlson ($p = 28$)	0.863	0.862
Labs ($p = 210$)	0.824	0.824
Demographic + Charlson + Labs ($p = 238$)	0.913	0.910

04/20 update

- Preliminary results on a small, **balanced** subsample (5000 cases, 5000 controls).
- Training set of 9000 IDs, test set of 1000 IDs.
- Imputed missing variables with median.
- **Sets of predictors:**
  - Demographic data includes race, age, weight, etc. (**Demographic**).
  - Charlson score inputs plus GERD diagnosis (**Charlson**).
  - Lab data (blood tests, FOBT) and colonoscopy data converted to longitudinal summaries (mean,max,min,...) (**Labs**).
  - Medication (H2R and PPI) converted to longitudinal summaries (mean,max,...) (**Meds**).
- Lab and medication data restricted to time interval from 'indexdate - 3 years' to 'indexdate - 1 year'.
- Logistic regression.

Predictors ($p = \#$ **of predictors)**	**Training AUC**	**Test AUC**
Demographic ($p = 10$)	0.814	0.818
Charlson ($p = 18$)	0.705	0.693
Meds ($p = 10$)	0.604	0.598
Demographic + Charlson + Labs + Meds ($p = 240$)	0.915	0.912

04/27 update

- Baseline logistic regression on entire sample (no blood labs, no medication data).
- Full sample contains $n = 6,649,108$ observations, with $n_{\text{control}} = 6,637,713$ and $n_{\text{case}} = 11,395$.
- Imputed numeric variables with medians, imputed smoking status at random, proportional to non-missing in sample (45% current, 41% former, 14% never).

Predictors (p = # of predictors)	**Training AUC**	**Test AUC**
Demographic ($p = 10$)	0.670	0.675
Charlson ($p = 18$)	0.684	0.705
Demographic + Charlson ($p = 28$)	0.760	0.770

Smoking status missingness.

In the entire sample:

Smoking Status	Current	Former	Never	Missing
Count	1696052	1521806	537727	2893523
Probability	0.26	0.23	0.08	0.44

Among cases:

Smoking Status	Current	Former	Never	Missing
Count	3901	3546	594	3354
Probability	0.34	0.31	0.05	0.29

Among controls:

Smoking Status	Current	Former	Never	Missing
Count	1692151	1518260	537133	2890169
Probability	0.25	0.23	0.08	0.44

05/04 update

- Preliminary results on a balanced **random sample** of 5000 cases and 5000 controls.
- Most missing variables imputed with their median, SmokeStatus imputed at random proportional to non-missing.
- Training on 9000 observations, testing on 1000 observations.
- Baseline logistic regression fit with subsets of predictors, and all predictors.
- Random forest fit with all predictors.

Model/predictors (p = # of predictors)	**Training AUC**	**Test AUC**
Logistic regression	-	-
Demographic ($p = 10$)	0.669	0.676
Charlson ($p = 18$)	0.689	0.659
Meds ($p = 10$)	0.515	0.513
Colonoscopies ($p = 2$)	0.513	0.502
Labs ($p = 200$)	0.671	0.621
All ($p = 240$)	**0.814**	**0.778**
Random forest	-	-
All ($p = 240$)	**0.985**	**0.844**

Random forest variable importance. Most important blood lab measurements (all means of measurements over the 2 year window):

1. Hematocrit value (CBC labs)
2. MCH value (CBC labs)
3. ALT value (LFT labs)
4. Alk. Phos. value (LFT labs)
5. AST value (LFT labs)
6. White blood cells (WBC) value (CBC labs)
7. Glucose value (BMP labs)

Next steps:

- Better approaches to impute missing variables (esp. SmokeStatus).
- Improved non-linear classifiers for subsample.
- Logistic regression baselines for the full sample of 6M observations.

05/11 update

- Baseline logistic regression on full sample (no medication data).
- $n = 6,649,108$ observations, with $n_{\text{control}} = 6,637,713$ and $n_{\text{case}} = 11,395$.
- Most missing variables imputed with their median, SmokeStatus imputed at random proportional to non-missing.
- For lab information, only means of each measurement, no longitudinal information so far.
- 05/27 update: Charlson and Demographic+Charlson updated to exclude Cancer and Metastatic Carcinoma.

Model/predictors (p = # of predictors)	**Training AUC**	**Test AUC**
Demographic ($p = 10$)	0.670	0.675
Charlson ($p = 16$)	0.684	0.705
Demographic + Charlson ($p = 26$)	0.760	0.770
Colonoscopies ($p = 2$)	0.511	0.504
Labs ($p = 34$)	0.604	0.601
Meds ($p = 10$)	0.523	0.512

Next steps:

- Process medication data.
- Approaches to impute missing variables (esp. SmokeStatus).
- Non-linear classifiers for subsample, plots/graphics to interpret predictor effects.

05/18 update

- Medication data processed for full sample, new line (red) added to last week's table of baseline logistic regression AUCs.
- Gradient boosting implemented for the random sample of 5000 cases and 5000 controls.

Model/predictors ($p =$ # of predictors)	**Training AUC**	**Test AUC**
Logistic regression	-	-
All ($p = 240$)	0.814	0.778
Random forest	-	-
All ($p = 240$)	**0.985**	0.844
Gradient boosting	-	-
All ($p = 240$), no interactions	0.900	0.847
All ($p = 240$), 2-way interactions	0.924	0.857
All ($p = 240$), 3-way interactions	0.937	**0.864**

Same gradient boosting results without Charlson score inputs or GERD at index:

Model/predictors ($p =$ # of predictors)	**Training AUC**	**Test AUC**
Gradient boosting	-	-
No Charlson inputs ($p = 222$), no interactions	0.881	0.811
No Charlson inputs ($p = 222$), 2-way interactions	0.898	0.820
No Charlson inputs ($p = 222$), 3-way interactions	0.909	0.817

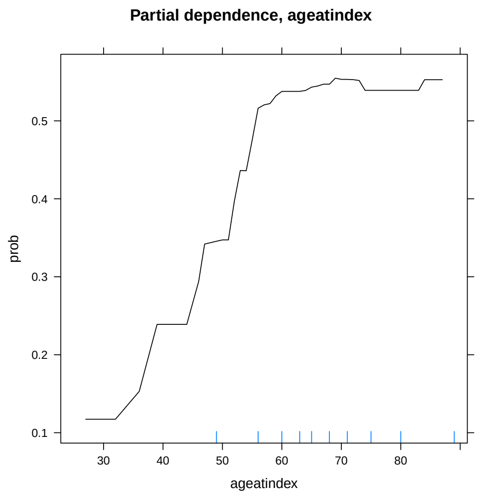
Next steps

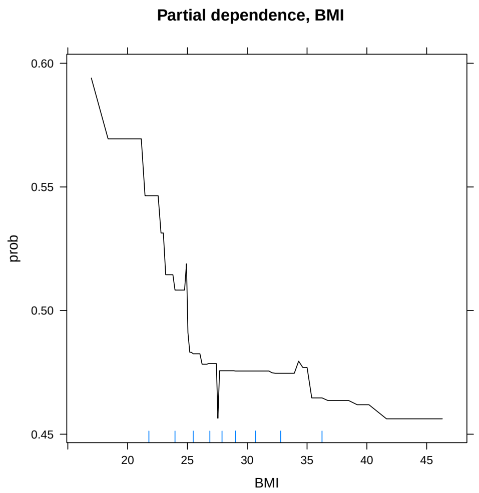
- Scaling up gradient boosting, dealing with unbalanced classes in full sample.
- Plots/graphics to interpret black box models.

05/25 update

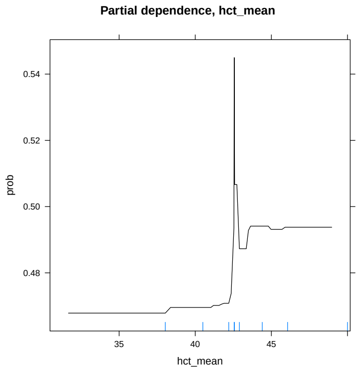
- Results now without Cancer and Metastatic Carcinoma inputs to Charlson score.
- Implemented scalable `xgboost` model, test AUC **0.860**.

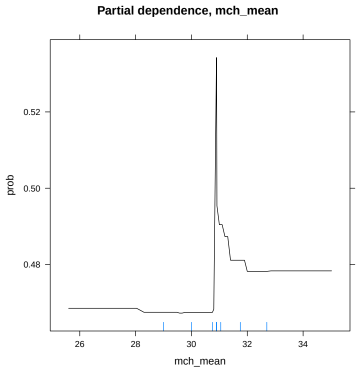
Partial dependence plots (from xgboost model). Demographic variables:

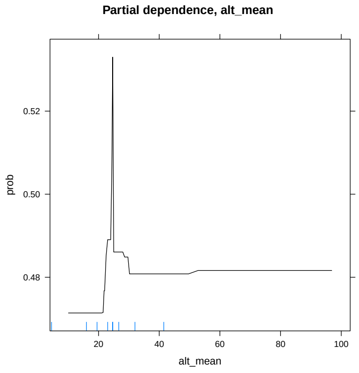


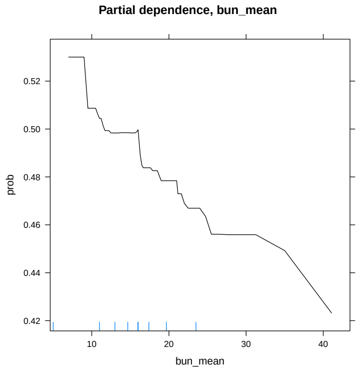


Most influential blood lab variables:









Missingness rates for blood lab variables (where 'missing' means zero measurements of that variable in the 2 year window):

Proportion missing	controls	cases
A1c_mean (A1C Labs)	0.51	0.52
bun_mean (BMP Labs)	0.17	0.30
hct_mean (CBC Labs)	0.20	0.50
CRP_mean (CRP Labs)	0.95	0.94
alkphos_mean (LFT Labs)	0.23	0.47
chol_mean (Lipid Labs)	0.19	0.32

08/20: now with the new data, 4 year window from index-1 to index-5

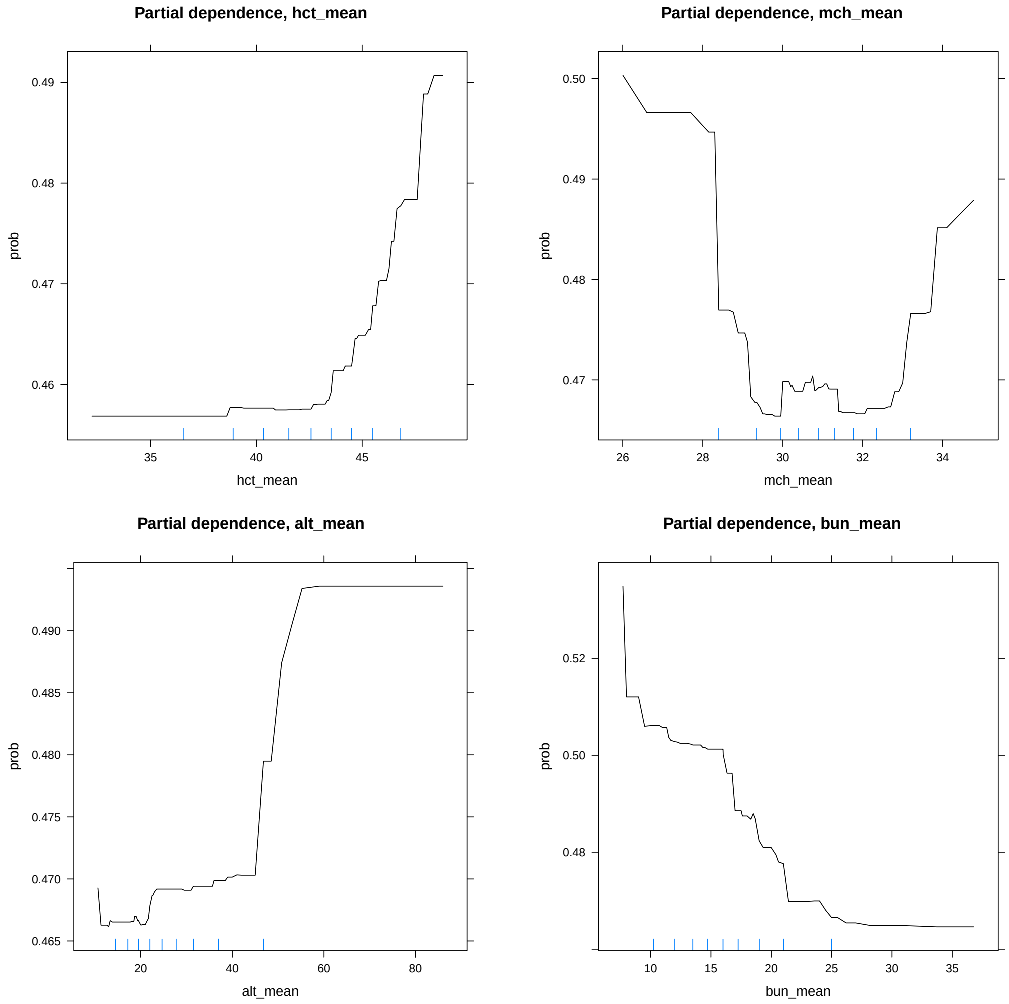
Proportion missing	controls	cases
A1c_mean (A1C Labs)	0.43	0.47
bun_mean (BMP Labs)	0.11	0.26
hct_mean (CBC Labs)	0.12	0.46
CRP_mean (CRP Labs)	0.93	0.91
alkphos_mean (LFT Labs)	0.15	0.42
chol_mean (Lipid Labs)	0.12	0.29

uniformly fewer missing, but rates are not cut in half, moreover heterogeneity of missingness between cases and controls is not fixed.

06/08 update

- Re-coded BMI/weight measurements?
- Re-run xgboost on balanced subsample with `NA`s, test AUC **0.880**, but it is using missing values to identify cases.
- Source of missingness? More detailed breakdown for LFT Labs.

Updated partial dependence plots for blood lab variables. Using new xgboost model on subsample.



Detailed breakdown of missingness for LFT labs. (Prediction window is second and third years before index)

Mean # of labs	controls	cases
First year before index	1.87	2.42
Second year before index	1.47	1.52
Third year before index	1.42	1.34

Detailed breakdown for the prediction window (index minus 3 years to index minus 1 year):

Proportion	controls	cases
At least 1 lab	0.83	0.69
At least 2 labs	0.67	0.59
At least 3 labs	0.45	0.44
At least 4 labs	0.30	0.31
At least 5 labs	0.18	0.21

Next Steps:

- Imputation with xgboost
- Implement xgboost for entire sample

06/22 update

- Set up code on GitLab (waljee-zhu-ml-projects/hosea-project).
- Received updated sample with recoded BMIs. BMIs are more likely to be missing for cases (14%) than controls (8%).
- Continuing to code imputation following Deng and Lumley (2021).

BMI comparison on the original sample data:

Original BMI	< 20	$\in (20, 25]$	$\in (25, 30]$	$\in (30, 35]$	$\in (35, 40]$	> 40
$\mathbb{P}(Control \mid BMI)$	0.9960	0.9978	0.9985	0.9985	0.9986	0.9986
$\mathbb{P}(Case \mid BMI)$	0.0040	0.0022	0.0015	0.0015	0.0014	0.0014

With the re-coded sample data:

New BMI	< 20	$\in (20, 25]$	$\in (25, 30]$	$\in (30, 35]$	$\in (35, 40]$	> 40
$\mathbb{P}(Control \mid BMI)$	0.9983	0.9985	0.9985	0.9984	0.9981	0.9979
$\mathbb{P}(Case \mid BMI)$	0.0017	0.0015	0.0015	0.0016	0.0019	0.0021

06/29 update

- Now using updated data (new BMIs, Charlson scores).
- For subsample, comparison of baseline logistic regression and xgboost models for different imputation approaches:

  1. **No imputation**: leave NAs in the data, learn a classification for subjects with missing data (only compatible with xgboost).
  2. **Median imputation**: impute all variables with median/most common class.
  3. **Regression imputation**: impute by (linear) regression of each predictor variable on the others (similar to MICE).
  4. **Random sample imputation**: impute each predictor by sampling at random from the non-missing entries.

AUC results for xgboost/logistic regression for different imputation approaches. Results still on balanced subsample.

Imputation method/model	**Training AUC**	**Test AUC**	**Test AUC (complete records)**
No imputation	-	-	-
xgboost	0.977	0.942	0.643
Median imputation	-	-	-
logistic regression	0.834	0.810	0.585
xgboost	0.970	0.857	0.623
Regression imputation	-	-	-
logistic regression	0.805	0.760	0.557
xgboost	0.948	0.815	0.705
Random sample imputation	-	-	-
logistic regression	0.724	0.689	0.724
xgboost	0.919	0.729	**0.745**

Regression imputation still has ‘regression to mean’ effect for imputed values, imputed values have smaller variance than the true values (with observation error). Maybe try a version that imputes a rank then resamples from observed values.

07/06 update

- Source of missingness in the data? About 85% of cases missing Charlson score inputs (compared to about 47% of controls).
- Missingness in test set/use case?
- Updated (in red) last weeks results with a test set of 1000 complete records (226 cases, 774 controls).
  - Poor generalization to complete cases for no imputation, median imputation implies that those models are fitting to the missingness, patterns won't continue to hold with fully observed data.
  - Regression imputation still has 'regression to mean' effect for imputed values, imputed values have smaller variance than the observed values (observation error).
  - Good generalization of random sample imputation implies that there is no bias introduced from training on data with missingness.

07/13 update

- Should we evaluate the model on complete records or missing/imputed records?
- Recall: previous results with an additional test set of 1000 complete records (226 cases, 774 controls).
  - New comparison for **multiple** random sample imputation: imputes by sampling from non-missing entries several times (reps) for each missing value, training on the multiple imputed records.
- Memory issues fitting xgboost on the full data, implementing "batched" training.

AUC's for xgboost/logistic regression for different imputation approaches. Results still on balanced subsample.

Imputation method/ prediction model	**Training AUC**	**Test AUC (imputed records)**	**Test AUC (complete records)**
No imputation	-	-	-
xgboost	0.977	0.942	0.643
Median imputation	-	-	-
logistic regression	0.834	0.810	0.585
xgboost	0.970	0.857	0.623
Regression imputation	-	-	-
logistic regression	0.805	0.760	0.557
xgboost	0.948	0.815	0.705
Random sample imputation	-	-	-
logistic regression	0.724	0.689	0.724
xgboost	0.919	0.729	0.745
Multiple rand. samp.	-	-	-
xgboost, 10 reps	0.909	0.742	0.781
xgboost, 20 reps	0.939	0.765	0.820
xgboost, 30 reps	0.948	0.768	0.781

Final two columns suggest two different evaluation metrics, ensuring the model performs well on **both** (A) new patients with imputed data and (B) new patients with fully observed blood labs, etc.

07/20 update

- Continuing to test/tune imputation approaches.
- Notes on decision curve analysis.

Imputation method/ prediction model	**Training AUC**	**Test AUC (imputed records)**	**Test AUC (complete records)**
Separate class	-	-	-
xgboost	0.977	0.942	0.643
Median	-	-	-
logistic regression	0.834	0.810	0.585
xgboost	0.970	0.857	0.623
Regression	-	-	-
logistic regression	0.805	0.760	0.557
xgboost	0.948	0.815	0.705
Regression v2	-	-	-
logistic regression	0.769	0.706	0.564
xgboost	0.928	0.722	0.764
Random sample	-	-	-
logistic regression	0.724	0.689	0.724
xgboost	0.919	0.729	0.756
Multiple rand. samp.	-	-	-
xgboost, 10 imputes × 50 trees	0.909	0.742	0.781
xgboost, 20 imputes × 25 trees	0.906	0.750	0.778
xgboost, 30 imputes × 20 trees	0.909	0.765	0.788
xgboost, 100 imputes × 5 trees	0.898	0.762	0.806

07/27 update

- Continuing to tweak regression imputation approaches.
- Comparing parameters/number of imputations needed for multiple sample imputation.
- Notes on decision curve analysis.

Next steps:

- Re-process new data when available

Imputation method/ prediction model	**Training AUC**	**Test AUC** **(imputed records)**	**Test AUC** **(complete records)**
Separate class	-	-	-
xgboost	0.977	0.942	0.643
Median	-	-	-
logistic regression	0.834	0.810	0.585
xgboost	0.970	0.857	0.623
Regression	-	-	-
logistic regression	0.759	0.704	0.657
xgboost	0.902	0.742	0.757
Single rand. samp.	-	-	-
logistic regression	0.724	0.689	0.724
xgboost	0.919	0.729	0.756
Multiple rand. samp.	-	-	-
xgboost, 10 imputes	0.909	0.742	0.781
xgboost, 20 imputes	0.906	0.750	0.778
xgboost, 30 imputes	0.909	0.765	0.788
xgboost, 100 imputes	0.898	0.762	0.806

08/03 update

- Received new data, loading/processing files
- Still issues with missing Charlson scores (0's vs NAs), relationship between CaseControl indicator and number of visits.

08/17 update

- Processed longitudinal summaries for BMP labs. Missingness rates down but still different for cases and controls. Continuing to process new blood lab data.
- Incidence rates for some example Charlson inputs: Peptic Ulcer Disease, Renal Disease, GERD. Full tables for all inputs are on the next page.

Variable **(ignoring NA's)**	Among cases	Among controls
pud	6.8% (120/1762)	3.8% (142821/3803451)
RD	11.8% (208/1762)	12.1% (460302/3803451)
GERD	-	-

Variable **(impute NA's as 0's)**	Among cases	Among controls
pud	1.1% (120/11395)	2.2% (142821/6637713)
RD	1.8% (208/11395)	6.9% (460302/6637713)
GERD	17.9% (2044/11395)	15.7% (1039206/6637713)

Cross-classified by number of BMP lab results (0, 1 or 2+):

Peptic Ulcer Disease, **controls**:

Number of BMP labs	0	1	NA
0	29.3%	0.9%	69.8%
1	32.2%	0.9%	66.8%
2+	61.5%	2.5%	36.0%

Peptic Ulcer Disease, **cases**:

Number of BMP labs	0	1	NA
0	8.6%	0.7%	90.8%
1	10.7%	0.4%	88.8%
2+	17.0%	1.3%	81.7%

Full tables for all Charlson indicators (red indicates these variables have been dropped from past models):

Variable **(ignoring NA's)**	Among cases	Among controls
CANCER	30.5% (538/1762)	18.3% (695558/3803451)
CHF	12.3% (217/1762)	11.8% (450090/3803451)
CTD	2.9% (51/1762)	3.0% (115492/3803451)
DEM	1.4% (24/1762)	2.3% (85704/3803451)
DIAB_C	15.5% (273/1762)	13.8% (525226/3803451)
HIV	0.3% (6/1762)	0.8% (30507/3803451)
MET_CAR	6.6% (117/1762)	1.3% (47696/3803451)
MLD	9.1% (160/1762)	8.0% (305462/3803451)
MSLD	0.7% (13/1762)	0.8% (29495/3803451)
PARA	1.9% (33/1762)	1.9% (72051/3803451)
RD	11.8% (208/1762)	12.1% (460302/3803451)
cd	16.3% (287/1762)	15.3% (582328/3803451)
copd	43.1% (759/1762)	35.9% (1365577/3803451)
diab_nc	44.6% (786/1762)	44.8% (1702629/3803451)
mi	9.1% (161/1762)	7.0% (264731/3803451)
pud	6.8% (120/1762)	3.8% (142821/3803451)
pvd	20.1% (354/1762)	16.0% (609425/3803451)
GERD	-	-

Variable **(impute NA's as 0's)**	Among cases	Among controls
CANCER	4.7% (538/11395)	10.5% (69558/6637713)
CHF	1.9% (217/11395)	6.8% (450090/6637713)
CTD	0.4% (51/11395)	1.7% (115492/6637713)
DEM	0.2% (24/11395)	1.3% (85704/6637713)
DIAB_C	2.4% (273/11395)	7.9% (525226/6637713)
HIV	0.1% (6/11395)	0.5% (30507/6637713)
MET_CAR	1.0% (117/11395)	0.7% (47696/6637713)
MLD	1.4% (160/11395)	4.6% (305462/6637713)
MSLD	0.1% (13/11395)	0.4% (29459/6637713)
PARA	0.3% (33/11395)	1.1% (72051/6637713)
RD	1.8% (208/11395)	6.9% (460302/6637713)
cd	2.5% (287/11395)	8.8% (582328/6637713)
copd	6.7% (759/11395)	20.6% (1365577/6637713)
diab_nc	6.9% (786/11395)	25.6% (1702629/6637713)
mi	1.4% (161/11365)	4.0% (264731/6637713)
pud	1.1% (120/11395)	2.2% (142821/6637713)
pvd	3.1% (354/11395)	9.2% (609425/6637713)
GERD	17.9% (2044/11395)	15.7% (1039206/6637713)

08/31 update

- Re-fit xgboost models on new subsampled data (compare to 07/27 results). $n_{\text{train}} = 9000$, $n_{\text{test}} = 1000$, $n_{\text{complete.test}} = 1000$.
- Compare (xgboost) model performance with different groups of variables included.
- Ongoing: a proxy for number of visits to better impute Charlson ICD codes?

Comparing different imputation approaches and prediction models:

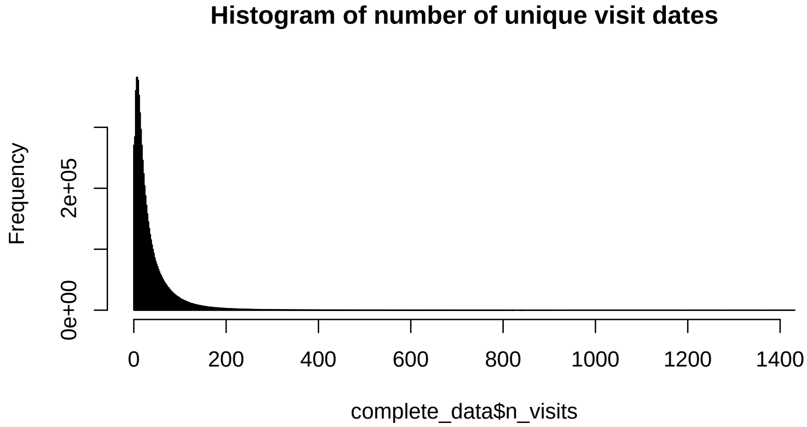
Imputation method/ prediction model	**Training AUC**	**Test AUC (imputed records)**	**Test AUC (complete records)**
Separate class	-	-	-
xgboost	0.985	0.962	0.582
Median	-	-	-
logistic regression	0.853	0.846	0.583
xgboost	0.963	0.911	0.618
Regression	-	-	-
logistic regression	0.772	0.726	0.651
xgboost	0.911	0.750	0.798
Single rand. samp.	-	-	-
logistic regression	0.739	0.695	0.703
xgboost	0.916	0.780	0.833
Multiple rand. samp.	-	-	-
xgboost, 10 imputes	0.942	0.824	0.885
xgboost, 20 imputes	0.935	0.820	0.872
xgboost, 30 imputes	0.943	0.810	0.879
xgboost, 100 imputes	0.931	0.824	0.891

Comparing different included variables (imputing with single random sampling, fit with xgboost):

Variables included	**Training AUC**	**Test AUC (imputed records)**	**Test AUC (complete records)**
Demographic ($p = 11$)	0.743	0.700	0.663
Charlson + GERD ($p = 16$)	0.600	0.542	0.530
Medications ($p = 10$)	0.594	0.501	0.539
Labs ($p = 202$)	0.903	0.687	0.829

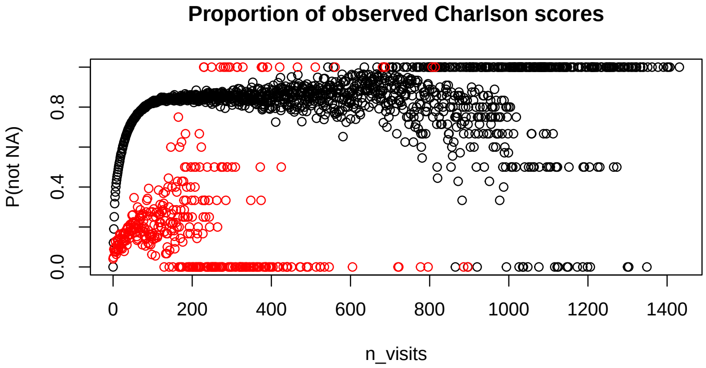
09/21 update

Overall missingness of Charlson score inputs. A histogram of the total number of visits in the 4-year prediction window:

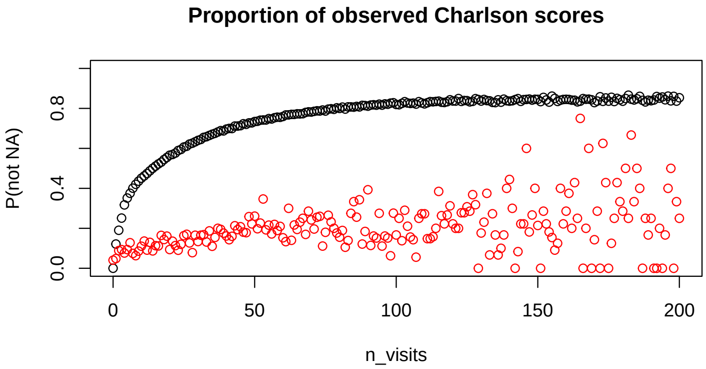


The median control has 22 visits, while the median case has 33 visits.

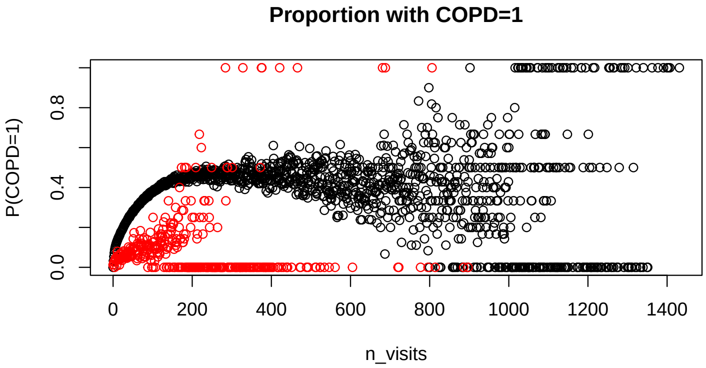
Proportion of observed Charlson scores plotted against number of visits, separating controls (black) and cases (red):



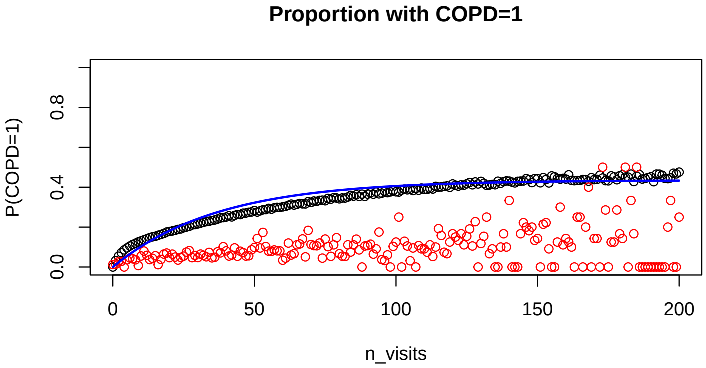
The same plot restricted to ≤ 200 visits.



Example: missingness of COPD. Proportion of patients coded COPD=1, plotted against number of visits, separating controls (black) and cases (red):



The same plot restricted to ≤ 200 visits. The blue line is a model-based estimate of $\mathbb{P}(\text{COPD} = 1|\text{n\_visits})$ using pooled case and control data.

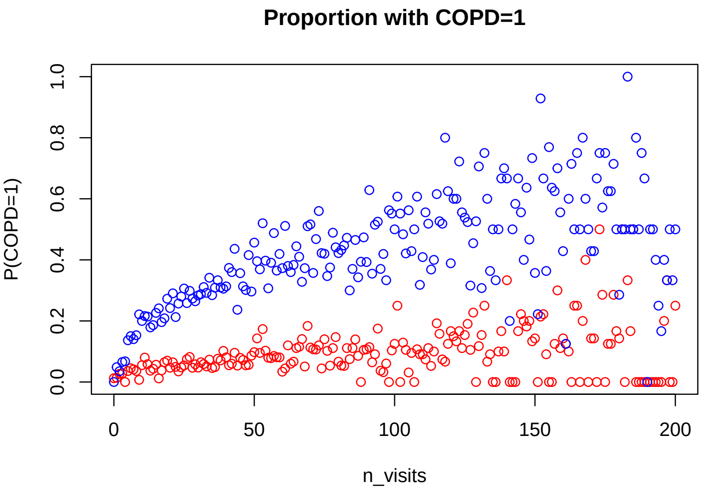


For cases only, created a new COPD indicator based on the ‘alldxscx’ table. For ICD9, 1 means you have a code ‘49x’, for ICD10, 1 means you have a code starting with ’J44’.

Cross-classification of old and new indicator for total of 11,395 cases (overall proportions in parentheses).

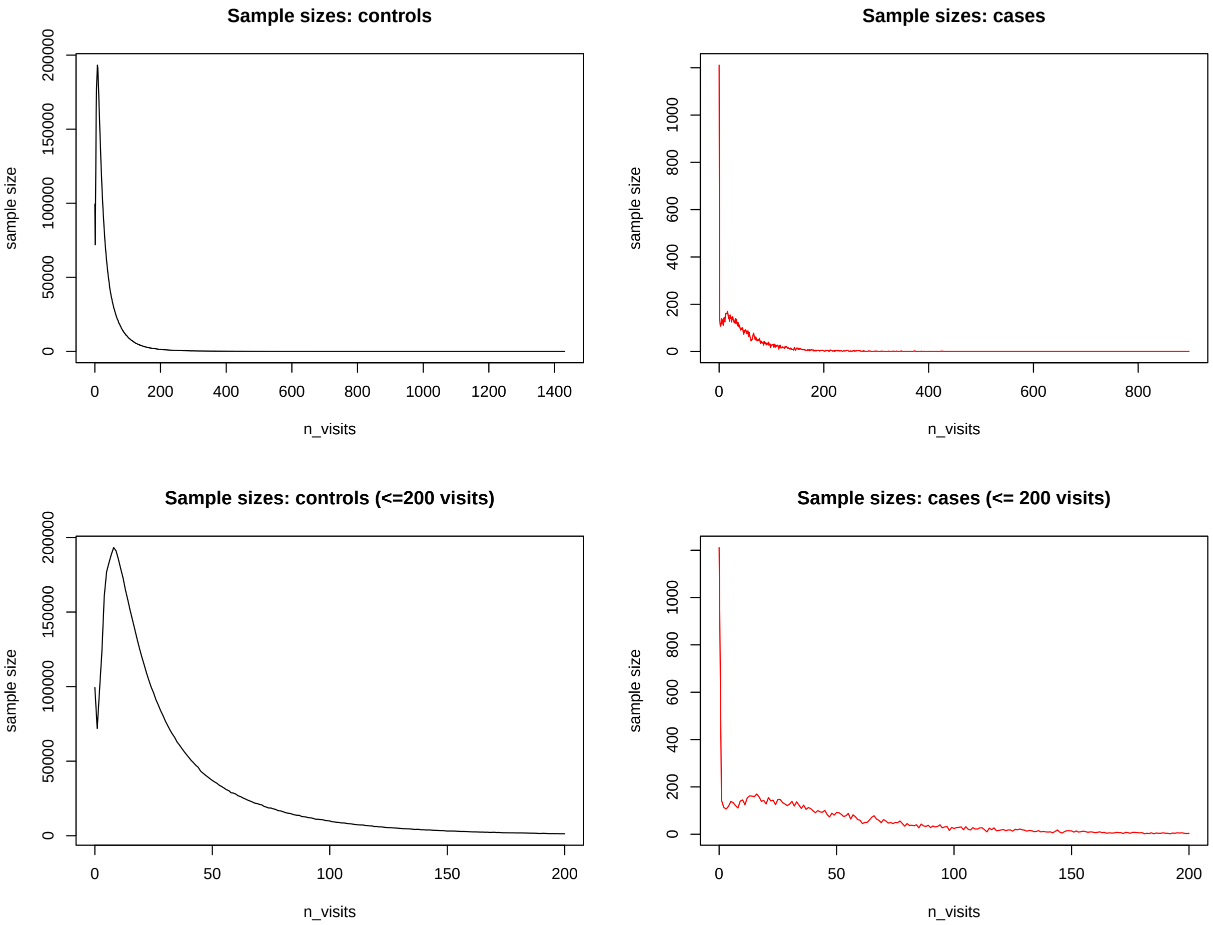
	New COPD=1	New COPD=0/NA
Old COPD=1	7845 (0.688)	2791 (0.245)
New COPD=0/NA	160 (0.014)	599 (0.053)

Proportion of patients with COPD=1 plotted against number of visits, restricted to ≤ 200 visits, with the old indicator in red and the new indicator in blue.



Next steps: similarly code a new indicator for controls, see if it changes results. Generalize to other Charlson inputs (or other entirely new ICD codes), and look at ways to impute by incorporating n_visits.

Sample sizes for controls and cases for calculating the proportions in the above plots.



09/28 update

- Implemented new train/valid/test scheme
  - Imputed records results are not exactly comparable
  - Complete records results are comparable
- Experimentation with regression imputation
- Experimentation with `xgboost` tuning parameters
- Currently working on multiple random sample with new train/valid/test scheme

	Training AUC	**Test AUC**	**Test AUC**
Imputation method		**(imputed records)**	**(complete records)**
Separate class	0.974 (0.985)	0.953 (0.962)	0.607 (0.582)
Median	0.979 (0.963)	0.939 (0.911)	0.649 (0.618)
Regression	0.952 (0.911)	0.739 (0.750)	0.770 (0.798)
1 rand. samp.	0.987 (0.916)	0.783 (0.780)	0.890 (0.833)
	10/04 update		
2 rand. samp.	1.000	0.801	0.918
5 rand. samp.	0.977	0.807	0.928
10 rand. samp.	1.000 (0.942)	0.818 (0.824)	0.915 (0.885)
20 rand. samp.	0.953 (0.935)	0.804 (0.820)	0.909 (0.872)
50 rand. samp.	0.978	0.814	0.912
100 rand. samp.	0.975 (0.931)	0.822 (0.824)	0.925 (0.891)

*all with `xgboost` prediction model; AUCs in parentheses denote previous results.

10/04 update

- Implemented new train/valid/test scheme with multiple imputation
- Updated table on previous page with result with multiple imputation
- Some improvement with a single random simpling imputation
- Some improvement compared to previous results, mostly for “complete records” test set
- Next step: utilize all of the data

10/12 update

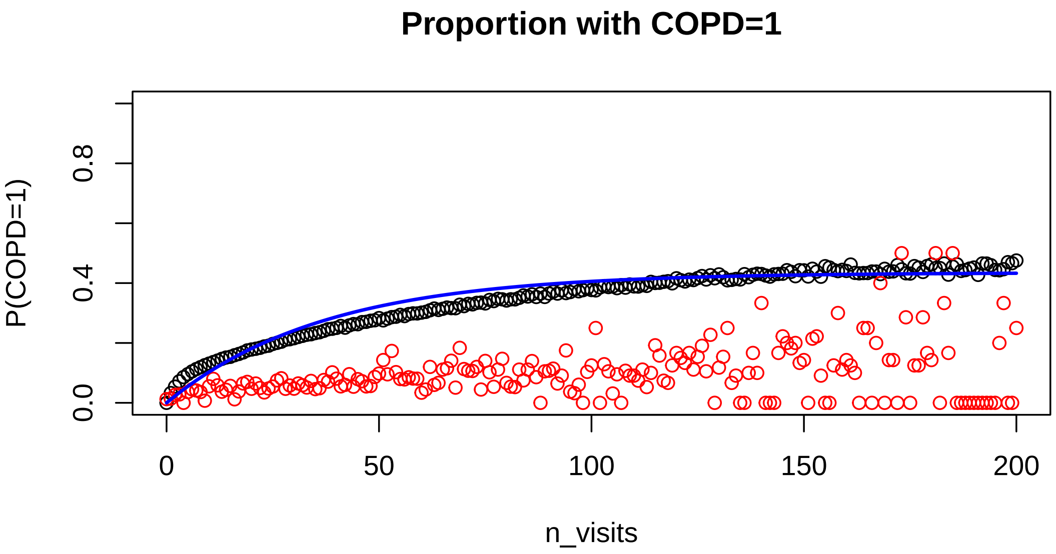
- Redo analysis with various Charlson variable treatments:
  - no Charlson
  - sampling from observed values (what was done previously)
  - imputing a probability using number of visit (see Peter's proposed Geometric model)
  - imputing by sampling using the fitted probability
- Results:
  - see table below
  - dropping Charlson leads to a small decrease in testing performance for both imputed and "complete" records
  - sampling with fitted probability has similar performance compared to the previous methods
  - imputing the fitted probability leads to a huge improvement in performance
- Currently examining where this large improvement comes from, still suspicious
- Try the same with multiple random samples
- Working on utilizing all data ...

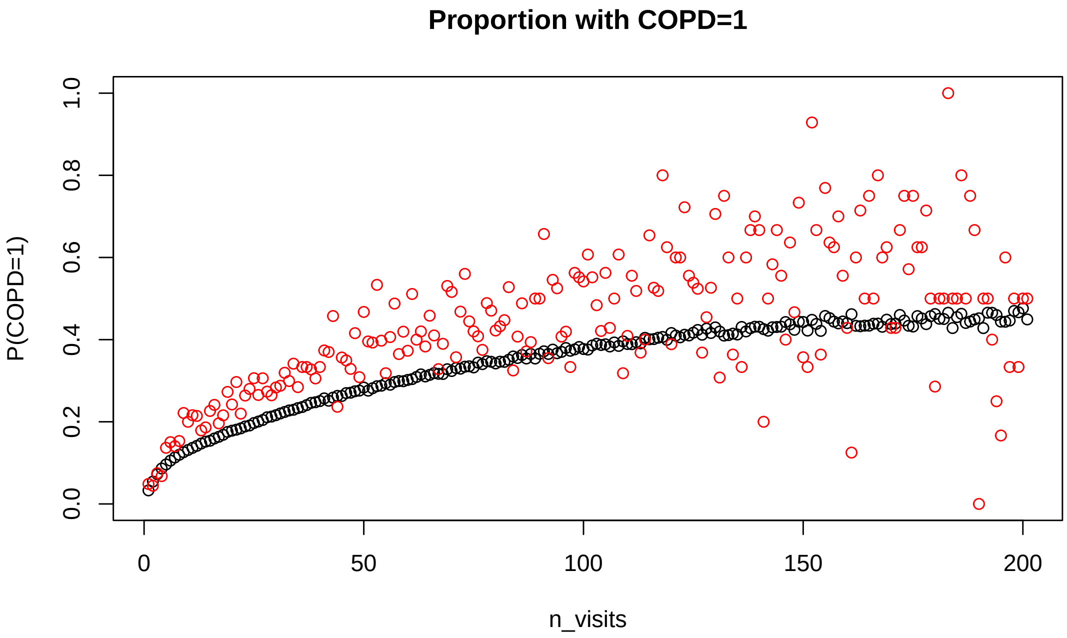
Imputation method	**Training AUC**	**Test AUC (imputed)**	**Test AUC (complete)**
Separate class	0.974	0.953	0.607
Median	0.990	0.941	0.674
Simple random sample			
no Charlson	0.988	0.771	0.873
random Charlson	0.999	0.786	0.905
proba. Charlson	1.000	0.906	0.975
proba.-weighted random Charlson	0.997	0.795	0.896

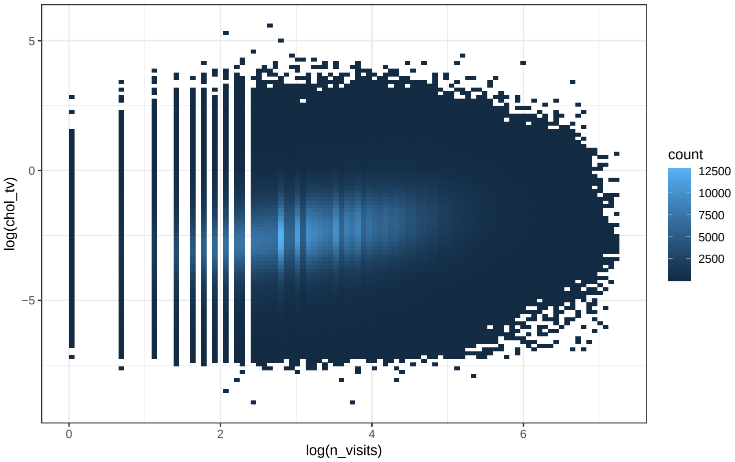
10/26 update

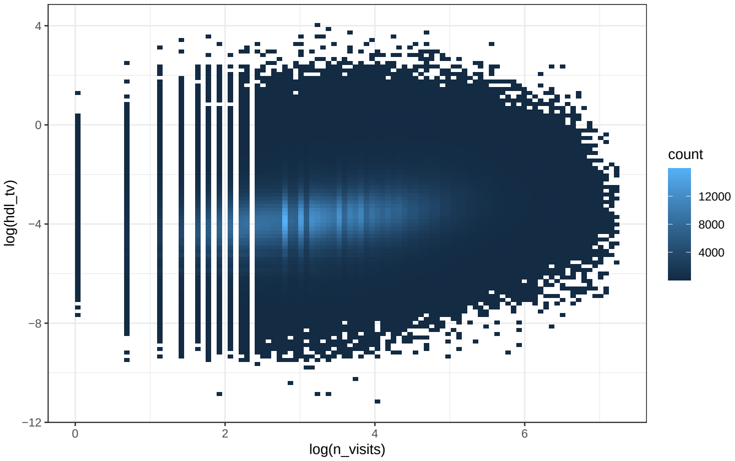
Some follow-ups:

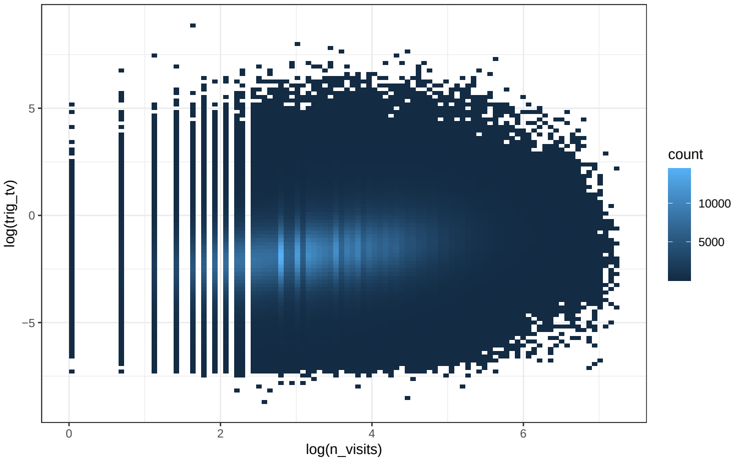
- Using number of visits to impute Charlson/GERD variables:
  - Imputing probabilities is not a good idea as it is easy to spot missing values (between 0 and 1) from observed values (0/1)
  - randomizing using the probability does not improve performance
  - preferable to have a unified treatment
- ICD codes reconding for cases comparing before and after
  - we no longer see the large discrepancy between cases and control
  - see COPD graphs below
- Does the "tv" lab summaries contain some information about the number of measurements?
  - there were some concern that variability would be associated with numer of measurements
  - see density plots for "chol", "hdl" and "trig" below
  - "tv" stands for "total variation" and is the average absolute change between measurement (standardized by time between measurements)
  - there does not seem to be strong associations




Proportion with COPD=1
P(COPD=1)
0.0
0.2
0.4
0.6
0.8
1.0
0
50
100
150
200
n_visits

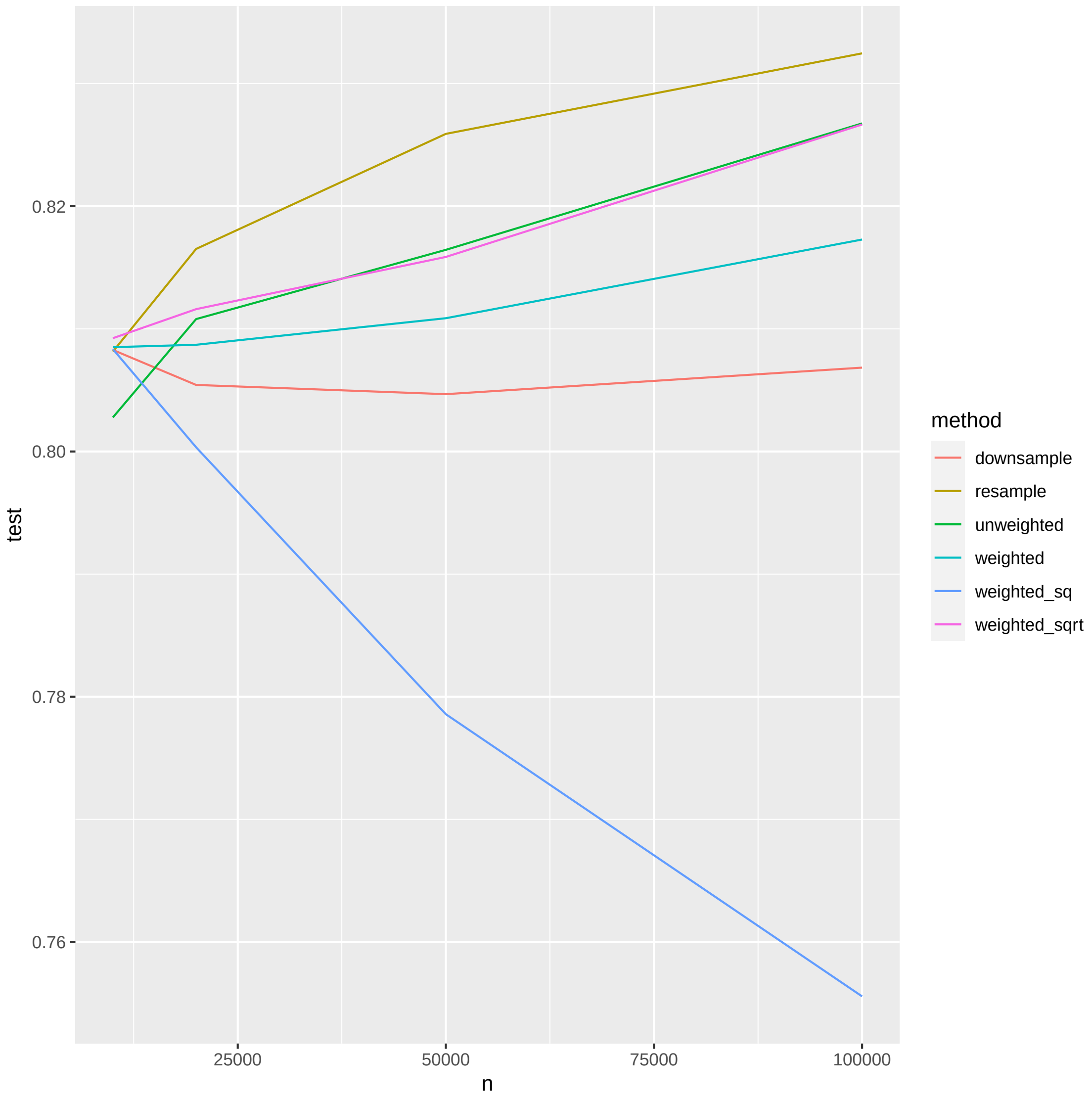

5
0
−5
log(chol_tv)
0
2
4
6
log(n_visits)
count
12500
10000
7500
5000
2500

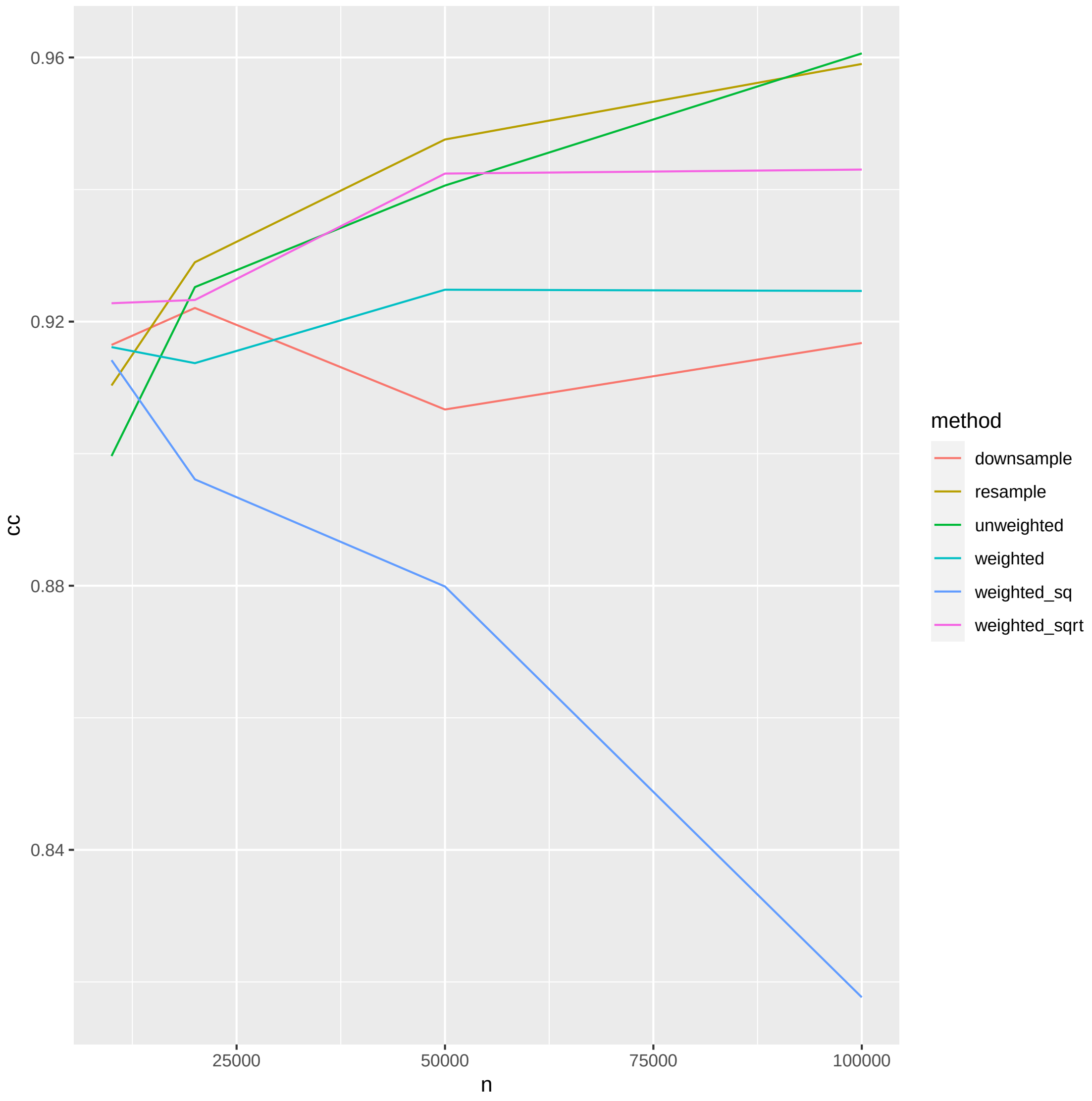

log(hdl_tv)
4
0
−4
−8
−12
0
2
4
6
log(n_visits)
count
12000
8000
4000


log(trig_tv)
5
0
−5
0
2
4
6
log(n_visits)
count
10000
5000

Using all data:

- Scalability study
- Sample 1M controls, keep all 11.4K cases (1M/11.4K)
- Train-test split 75-25 stratified
  - Fixed test set (250K/2.8K)
  - Training using increasing subsets of (750K/8.5K)
- Working training set consisting of downsampling the controls
  - n =10K, 20K, 50K, 100K
  - keeping all 8.5K cases
  - as n increases, the training set becomes more and more unbalanced
  - n =10K is close to balanced
- Validation set: stratified 2-1
  - Model/imputation trained on 67%, validation/model selection on 33%
- Extra testing set: "complete cases" (same as before, 710/290)
  - small, with possible overlap
  - not a super reliable testing set
- Methods (e.g. with 100K/8.5K → 66.7K/5.7K)
  - Unweighted: do nothing special
  - Downsample: sample controls to get a balance training set (5.7K/5.7K)
  - Resample: resample cases to get a balance training set (66.7K/66.7K)
  - Weighted: cases' loss reweighted by $w = 66.7/5.7$
  - Weighted_ sqrt: by $\sqrt{(}w)$
  - Weighted_ sq: by w^2


test
0.82
0.80
0.78
0.76
25000
50000
75000
100000
n
method
downsample
resample
unweighted
weighted
weighted_sq
weighted_sqrt


0.96
0.92
0.88
0.84
cc
25000
50000
75000
100000
n
method
downsample
resample
unweighted
weighted
weighted_sq
weighted_sqrt

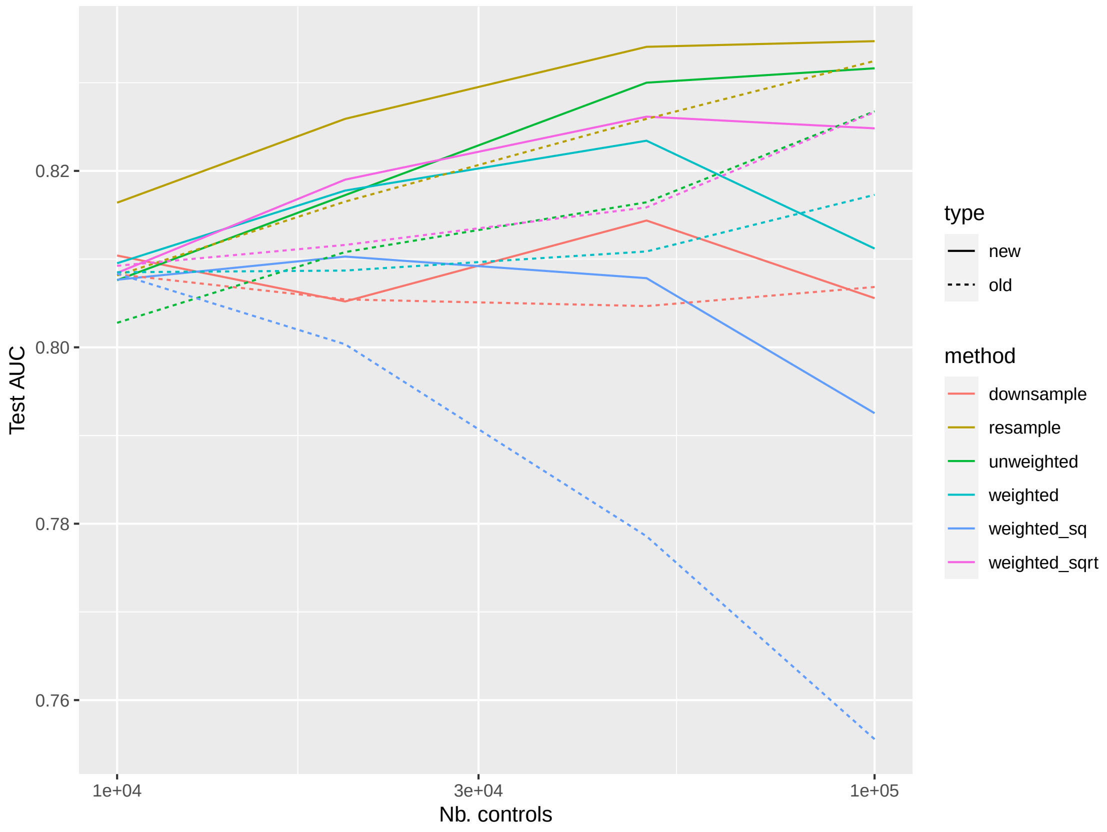
11/02 update

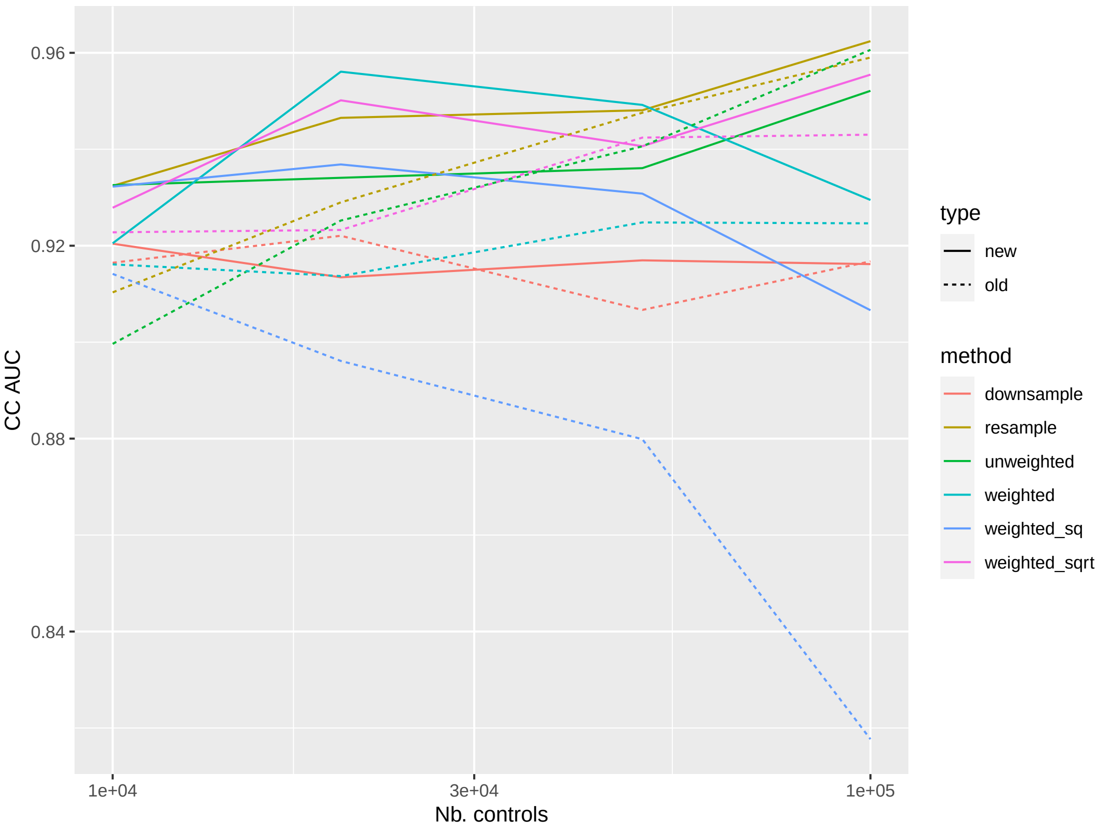
Tuning

- subsampling each iteration, learning rate, tree depth
- Some improvement is still possible
- Dependent on the training set size
  - Tuned on small smaple size, improvments don't necessarly carry to larger sample size
  - prohibitive to tune on large dataset

Training set size

- Training becomes *very* slow
- Early results show improvement beyond 100K
- Exploring ways to speed up training (lightgbm, catboost)


Test AUC
0.82
0.80
0.78
0.76
1e+04
3e+04
1e+05
Nb. controls
type
new
old
method
downsample
resample
unweighted
weighted
weighted_sq
weighted_sqrt


CC AUC
0.96
0.92
0.88
0.84
1e+04
3e+04
1e+05
Nb. controls
type
new
old
method
downsample
resample
unweighted
weighted
weighted_sq
weighted_sqrt

11/09 update

Training set size

- Added results for $> 100K$ controls
- Goal:
  - Harness all data
  - Study how the increasing case/control imbalance affects results
- Similar train/valid/test split as before
  - 2M control + 11K cases (increased from 1M previously to allow the 1M case)
  - All splits are stratified (cases/control proportion preserved)
  - test: 25%, train+valid: 75%
  - N: number of controls in train+valid
  - subsample train+valid controls down to N
  - train: 67%, valid: 33%
  - NB: test set is fixed with N, validation set increases with N
- Imputation:
  - Simple random sampling from the training set into validation and testing sets
- `xgboost` parameters:
  - Default values, except:
  - `subsample = 0.5`
  - `max_depth = 5`
  - `eta = 0.5` (stepsize)
  - NB: these values are represented with vertical dashed lines in the following "Tuning" section
- Methods:

  downsample sample controls down to the number of cases

  resample sample (w/ replacement) the controls up to the number of controls (note that controls appearing more than once get different imputation)

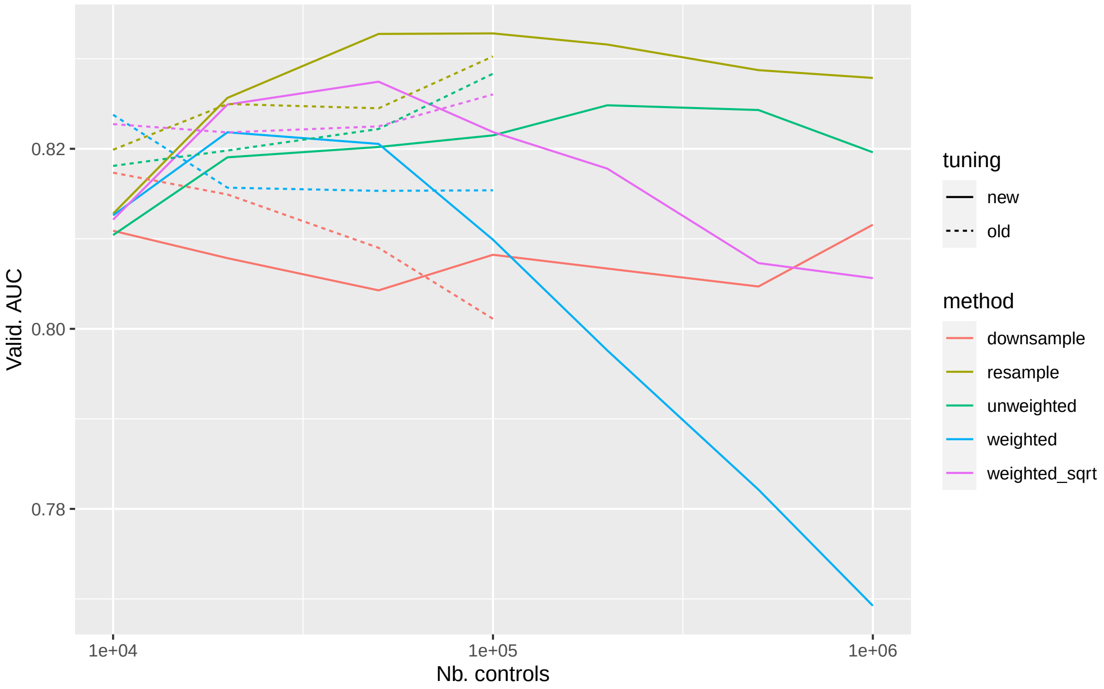
  unweighted no resample, no rewighing

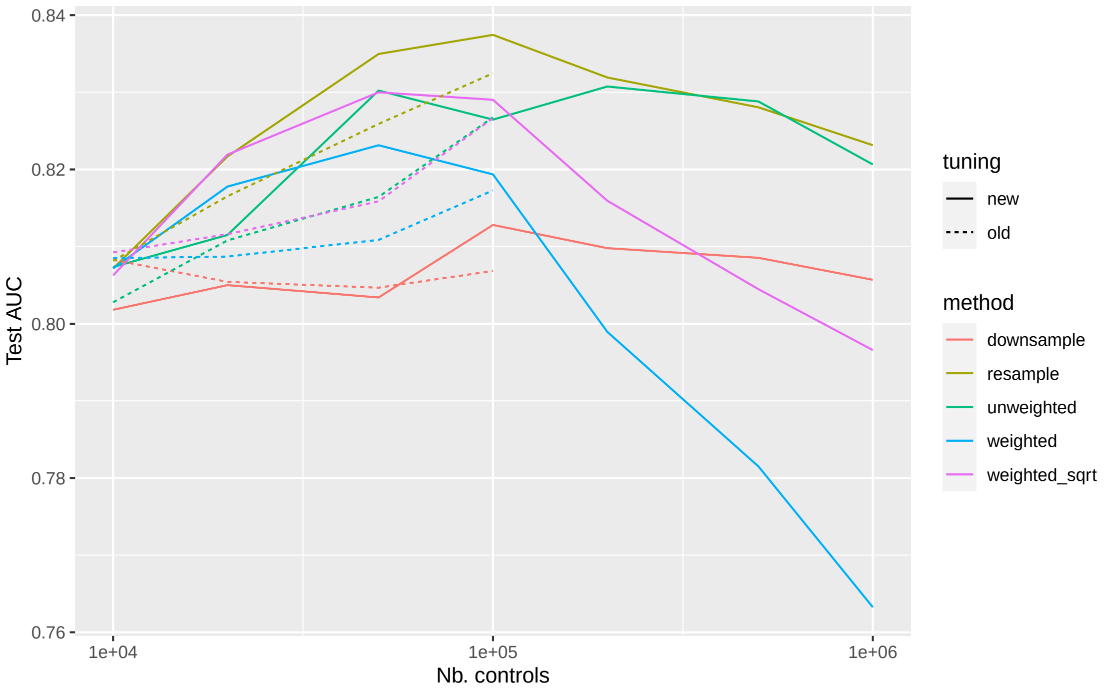
  weighted no resampling, cases upweighted by $n_{\text{controls}}/n_{\text{cases}}$

  weighted_sqrt no resampling, cases upweighted by $\sqrt{n_{\text{controls}}/n_{\text{cases}}}$

weighted_sq no resampling, cases upweighted by $(n_{\text{controls}}/n_{\text{cases}})^2$ (dropped from graph because much worse values and changed the scale too much to see others well)

- Plots:
  - y-axis: Test set AUC and Validation set AUC
  - x-axis: Nb. of controls before train/valid split
  - colors: which method is used to deal with imbalance
  - linetype: new vs. old tuning. The new tuning was performed manually with 100K controls; the old tuning with 10K controls. For more on tuning, see next section.
- Results and analysis:
  - Single repetition, so there is some variablity due to the sampling. To get a sense of the scale of the variability, the "downsample" method should be the same across the range. I might do multiple repititions in the future, but this gets very long for large N, so I am avoiding it for now.
  - The "resampling" method seems to be the best across the board
  - The "unweighted" method seems more stable?
  - Not much to gain beyond 50K-100K (decrease? within variability?). The best tuning parameters might not be the same for larger datasets, as is apparent between old/new tuning curves.




0.84
0.82
0.80
0.78
0.76
Test AUC
1e+04
1e+05
1e+06
Nb. controls
tuning
new
old
method
downsample
resample
unweighted
weighted
weighted_sqrt

Tuning

- Protocol:
  - Same as above, except:
    - Fix Nb. controls to $N = 100K$
    - Use “resample” method only
  - Fix all parameters to their default value stated above (and blotted with a vertical dashed line), vary one parameter along a plausible range
- Parameters:

  max_depth Control the complexity of the tree added at each iteration (deeper trees have more explanatory power, but can overfit)

  subsample Determines what percetage of the training set is used to fit the next tree (more subsampling prevents overfitting but slows down fitting)

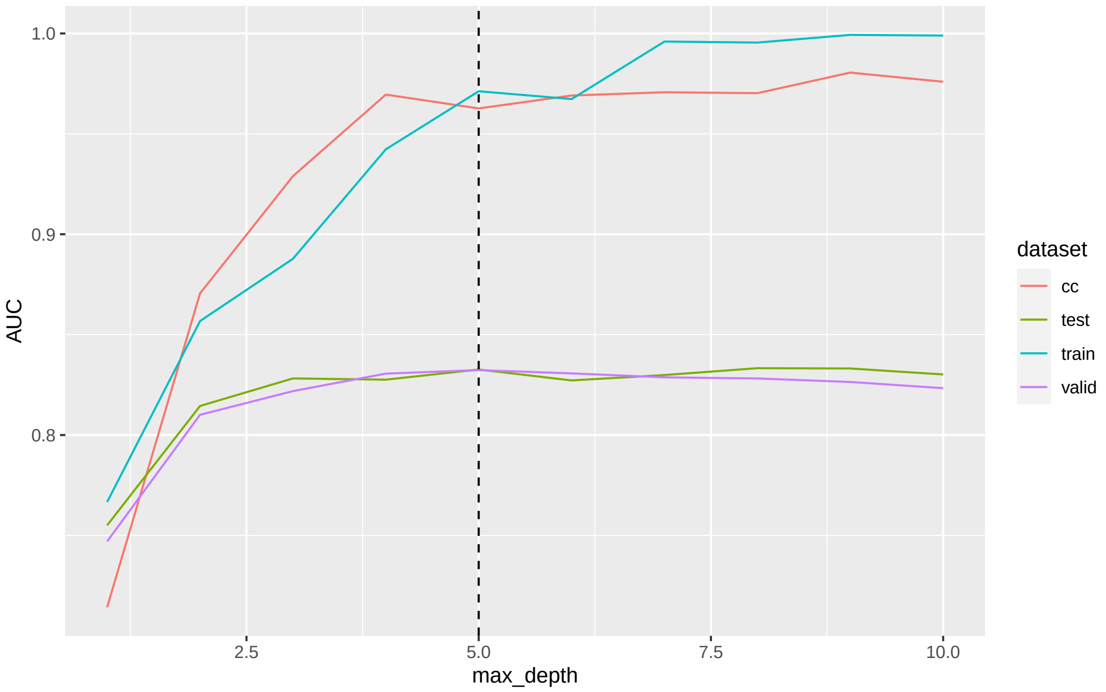
  eta (step-size) controls how much the new tree contributes to the model (small step-size prevents overfitting, but slows down fitting)

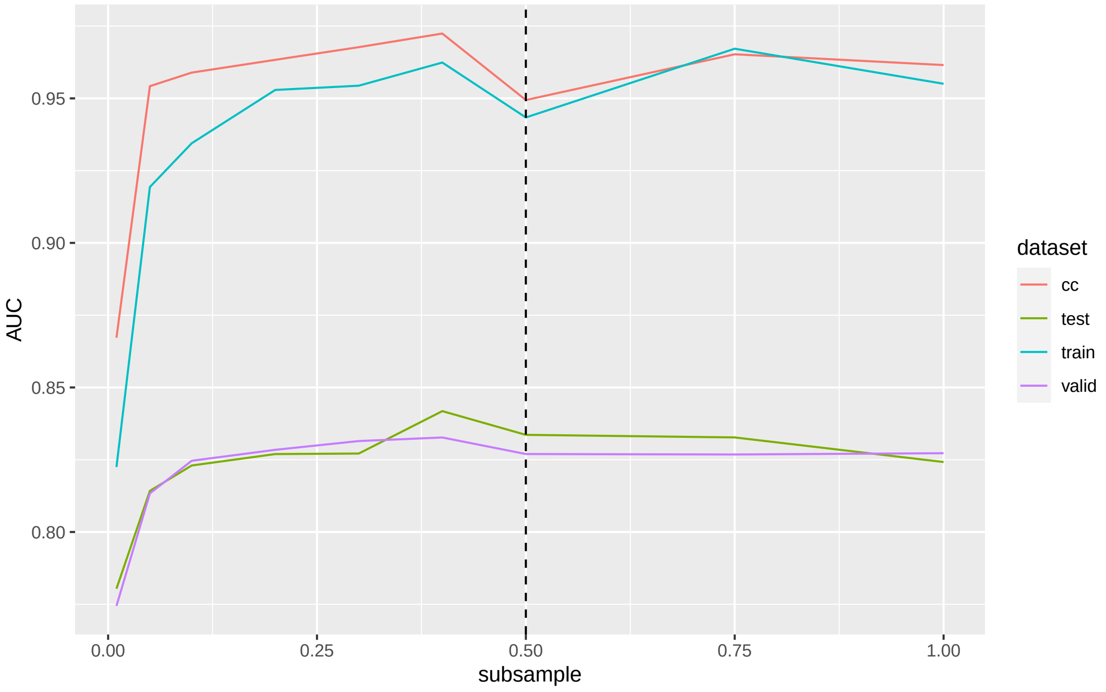
- Results and analysis:

  max_depth Above depth 4-5, not much variation on valid/test AUCs; some improvment on train/cc AUCs

  subsample Except for small proportions, not much variation for all 4 datasets; best seems to be around 0.4-0.75

  eta waiting for results ...


AUC
1.0
0.9
0.8
2.5
5.0
7.5
10.0
max_depth
dataset
cc
test
train
valid


AUC
0.95
0.90
0.85
0.80
0.00
0.25
0.50
0.75
1.00
subsample
dataset
cc
test
train
valid