Predicting Ready Graduates

Alexander Poon
Tennessee Department of Education

May 22, 2018

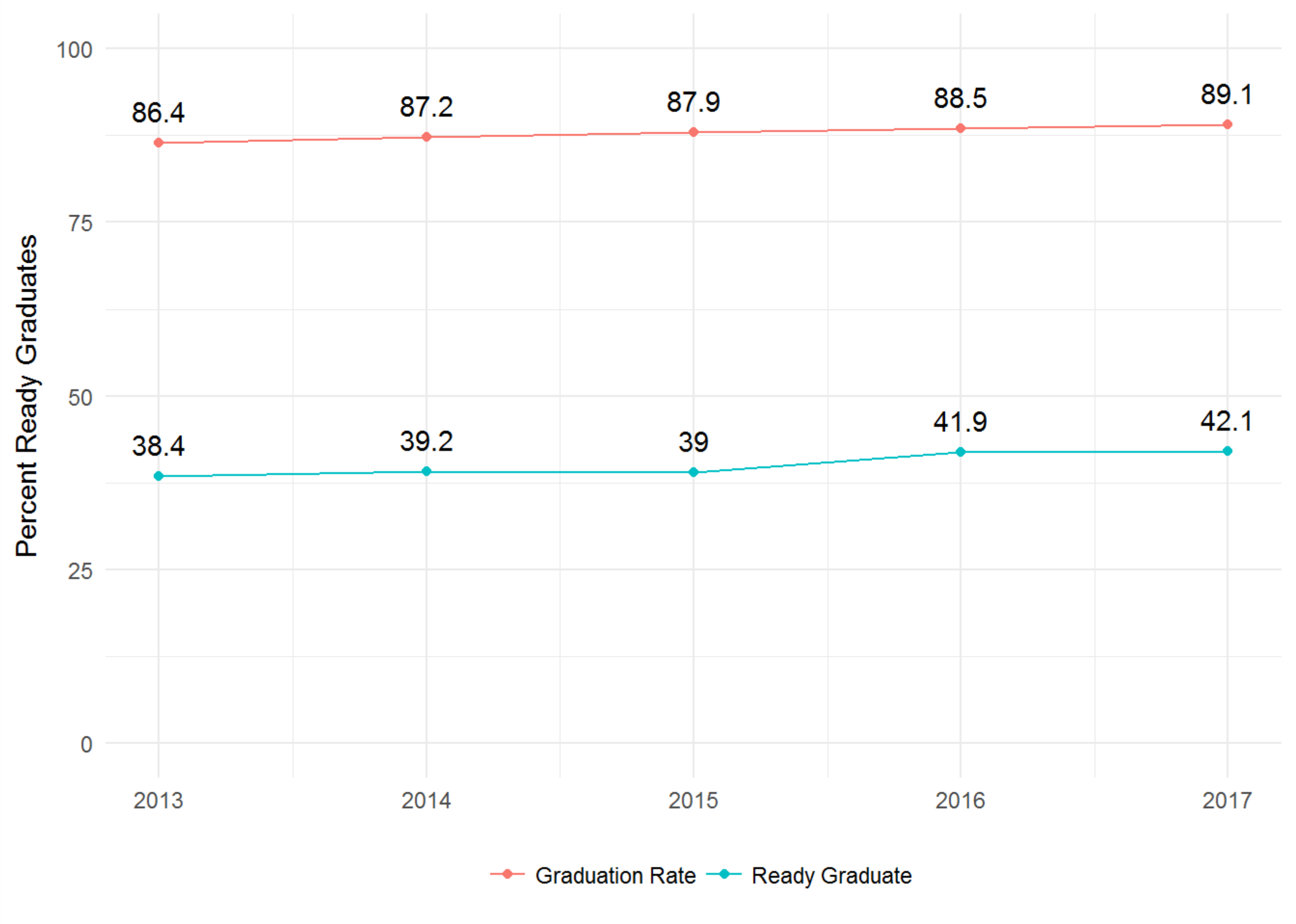
Motivation

- Bridge to Postsecondary priority to ensure that high school graduates are ready to pursue a postsecondary credential.
- Addition of ready graduates (on-time grad w/ 21+ ACT composite score) indicator to accountability.
- Following Dropout Early Warning Systems in WI and Chicago, looking to create a system to predict ready graduates.

Ready Graduation in TN

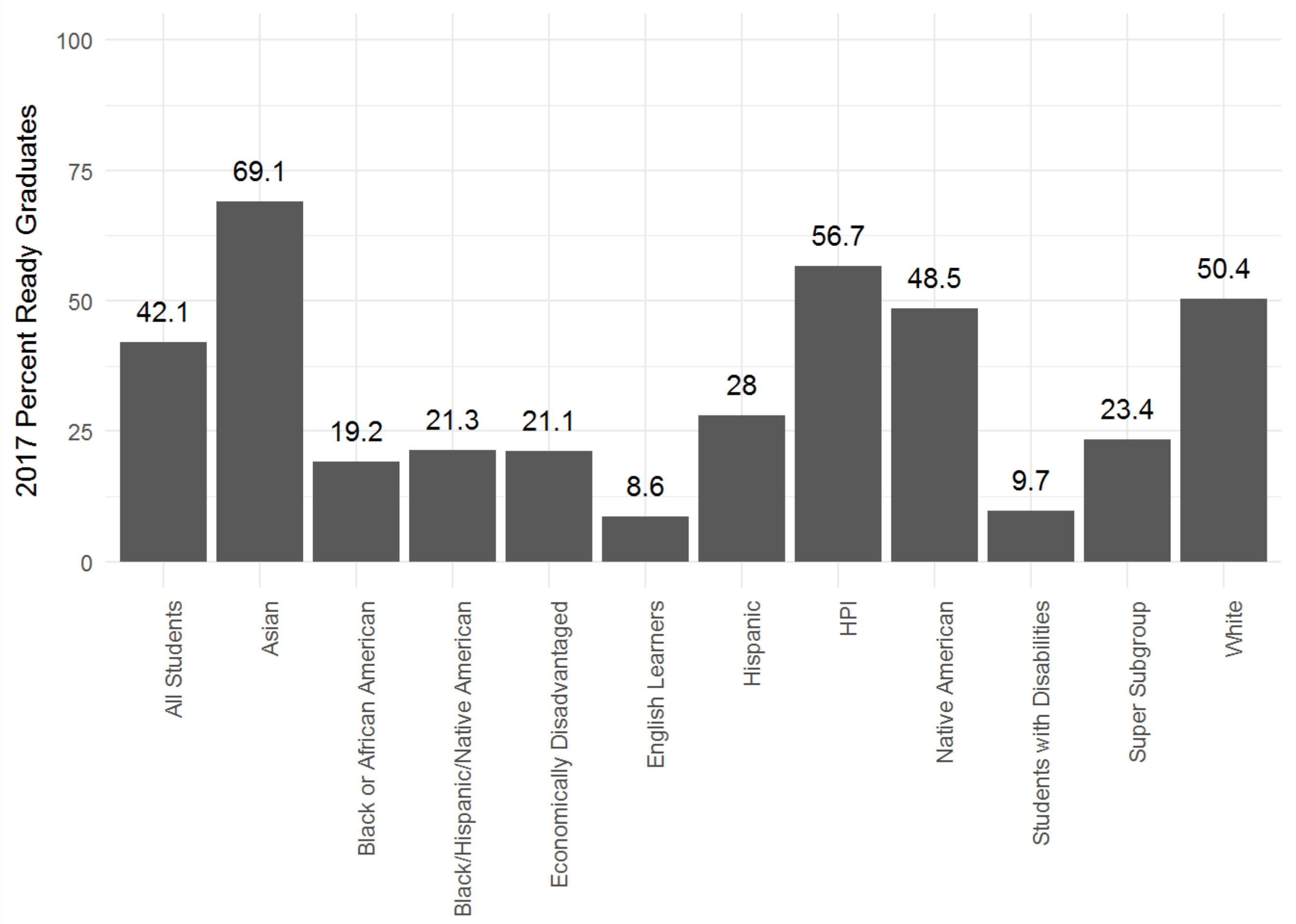
Graduation Rates and Ready Graduates

Ready Graduation rates lag behind Graduation Rates in Tennessee:



Ready Graduation by Student Groups

There are wide discrepancies across student groups in ready graduation rates:



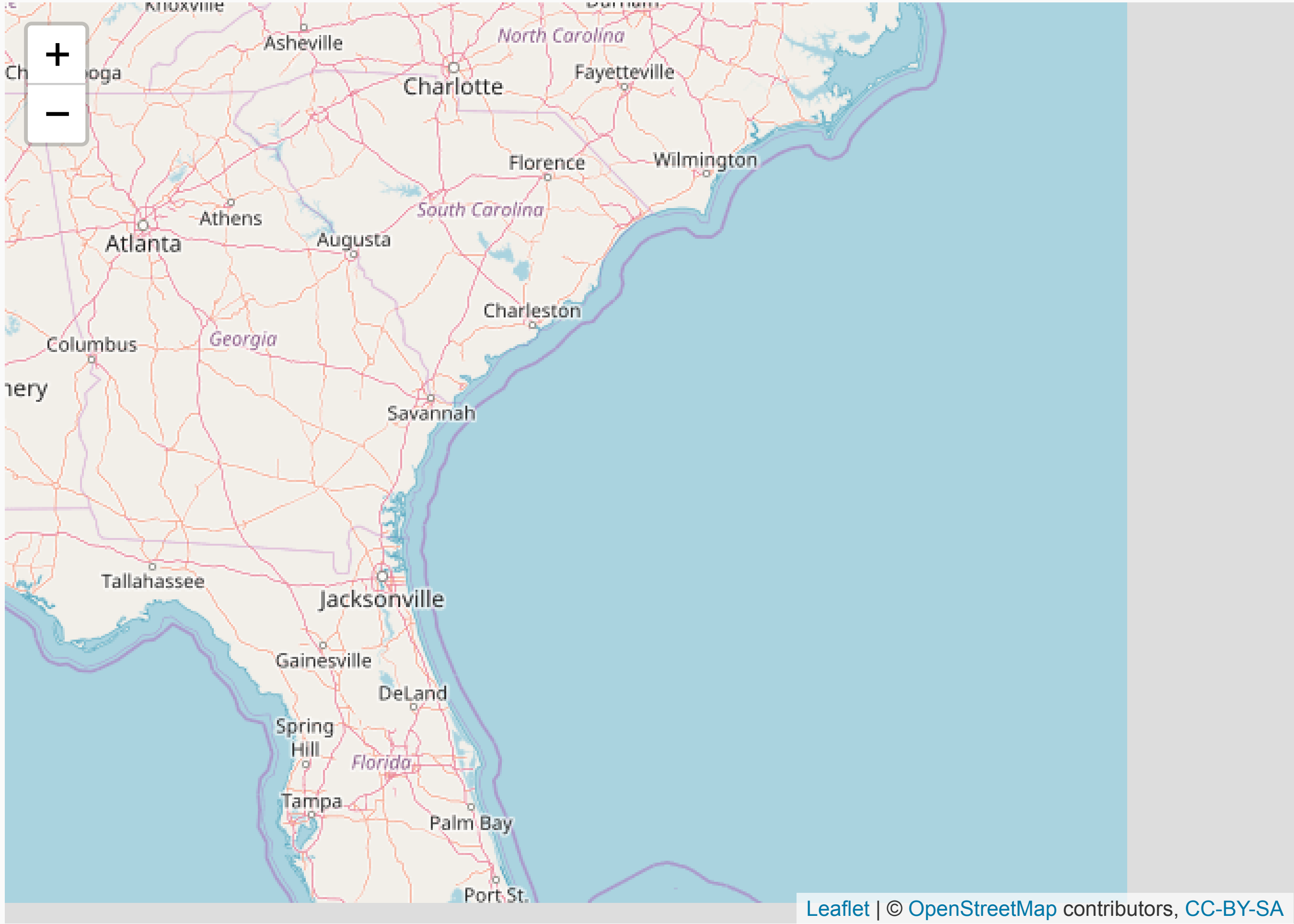
Ready Graduation by District

Districts with highest and lowest percentage of ready graduates:

District	% Ready Grads
796	82.5
940	79.8
795	75.7
52	67.8
793	65.7
12	64.9
822	59.1
901	57.1
231	54.9
920	54.6

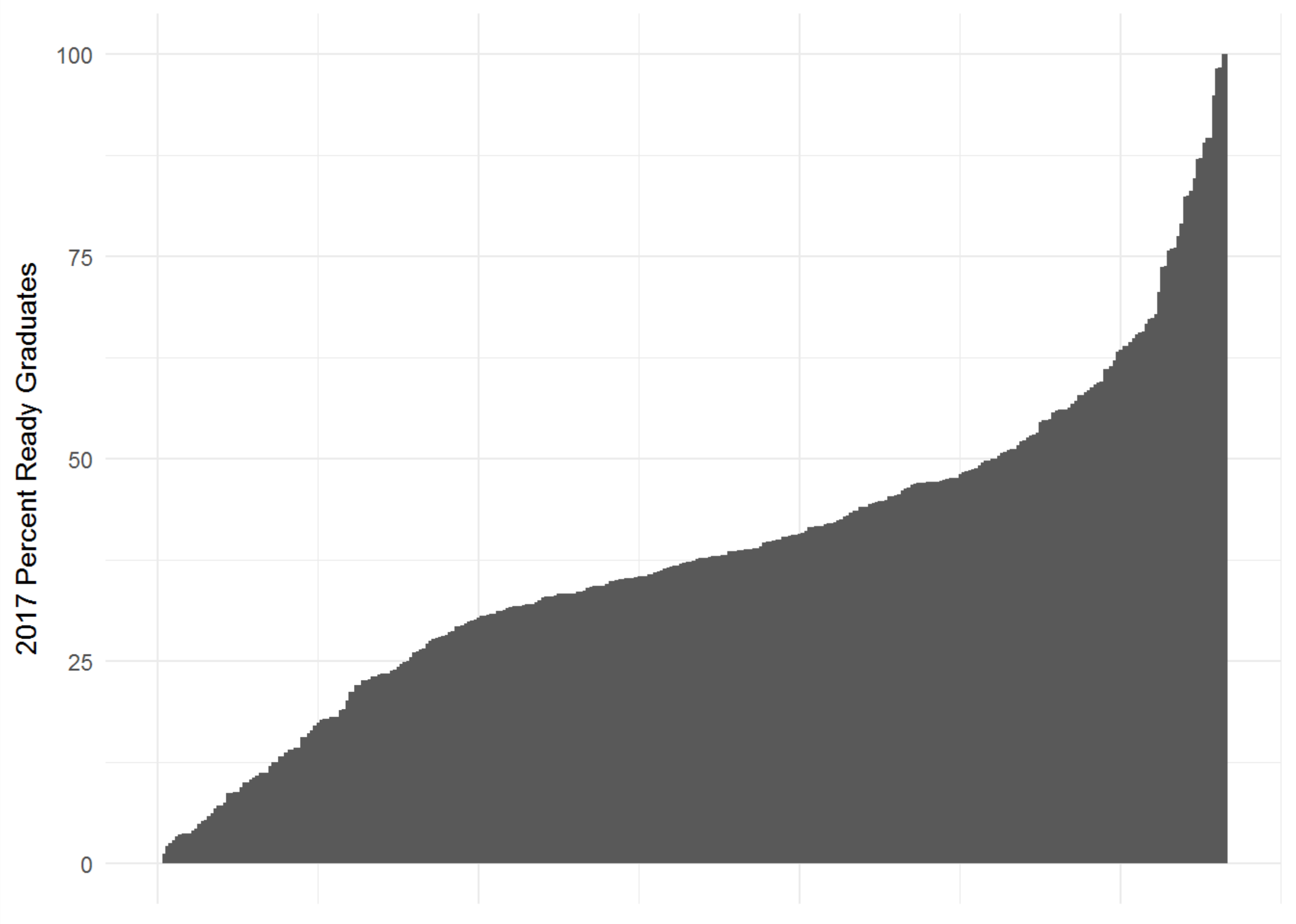
District	% Ready Grads
792	23.2
760	23.1
490	22.8
380	22.7
480	22.0
92	22.0
70	21.9
271	16.4
240	11.1
985	6.2

Ready Graduation by District



Ready Graduation by School

Tennessee has high schools at all levels of ready graduation:



Predicting Ready Graduation

Machine Learning

From Wikipedia:

[A] field of computer science that uses statistical techniques to give computer systems the ability to "learn" (i.e., progressively improve performance on a specific task) with data, without being explicitly programmed.

Machine Learning

The 'task' we want to 'learn' is to identify ready graduates based on information about students in 6th/7th/8th grades.

Absences	**Math Score**	**ELA Score**	**...**
1	Below Basic	Basic	...
20	Basic	Proficient	...
5	Basic	Basic	...
0	Advanced	Proficient	...
2	Advanced	Advanced	...
10	Basic	Advanced	...
0	Proficient	Proficient	...
3	Below Basic	Below Basic	...
10	Basic	Advanced	...
45	Proficient	Advanced	...

Ready Graduate
not ready
not ready
not ready
ready
ready
not ready
ready
not ready
ready
not ready

Machine Learning

To do this, I look at which 6th/7th/8th graders have ended up as ready graduates in the past and use this information to predict ready graduation for future cohorts.

I 'train' a model with the 2015 graduating cohort and predict on the 2016 graduating cohort so that I can check prediction accuracy.

Predictors

I use the following predictors for students in 6th, 7th, and 8th grades:

- TCAP Scores
- Disciplinary Issues (Suspensions/Expulsions/Reasons)
- Absenteeism (# days)
- Mobility (# of schools enrolled)
- School aggregates of above

Predictors

I use the following predictors for students in 6th, 7th, and 8th grades:

- TCAP Scores
- Disciplinary Issues (Suspensions/Expulsions/Reasons)
- Absenteeism (# days)
- Mobility (# of schools enrolled)
- School aggregates of above

Also considered:

- Course Grades
- Demographics

Predictor Considerations

- Using multiple years of data results in missingness as students enter and leave the state
- Need to have complete coverage for the state (grades)
- Need to have data going back at least 5 years (grades)
- Don't want to bias against certain groups of students (demographics)

Models

Model Example

Fitting a logistic regression model:

```r
# Predictors
predictors <- c("n_absences", "enrollments", "E", "I", "R", "S", "assault", "weapons", "theft_vandalism",
    "sexual_assault_harassment", "drugs_alcohol", "threat", "school_rules", "bullying", "fighting",
    "scale_score_mt", "scale_score_mt_sq", "scale_score_rd", "scale_score_rd_sq",
    "school_scale_score_mt", "school_scale_score_rd", "school_chronic_abs")

# Train a model on one cohort and test predictions on second
c(train_set, test_set) %<-% split(prediction_data_8, prediction_data_8$cohort)

train_x <- train_set[predictors]
train_y <- train_set$ready_grad

# Fit a model on train set
model_logistic <- train(x = train_x, y = train_y,
    method = "glm", family = "binomial")
```

Extracting Predictions from a Model

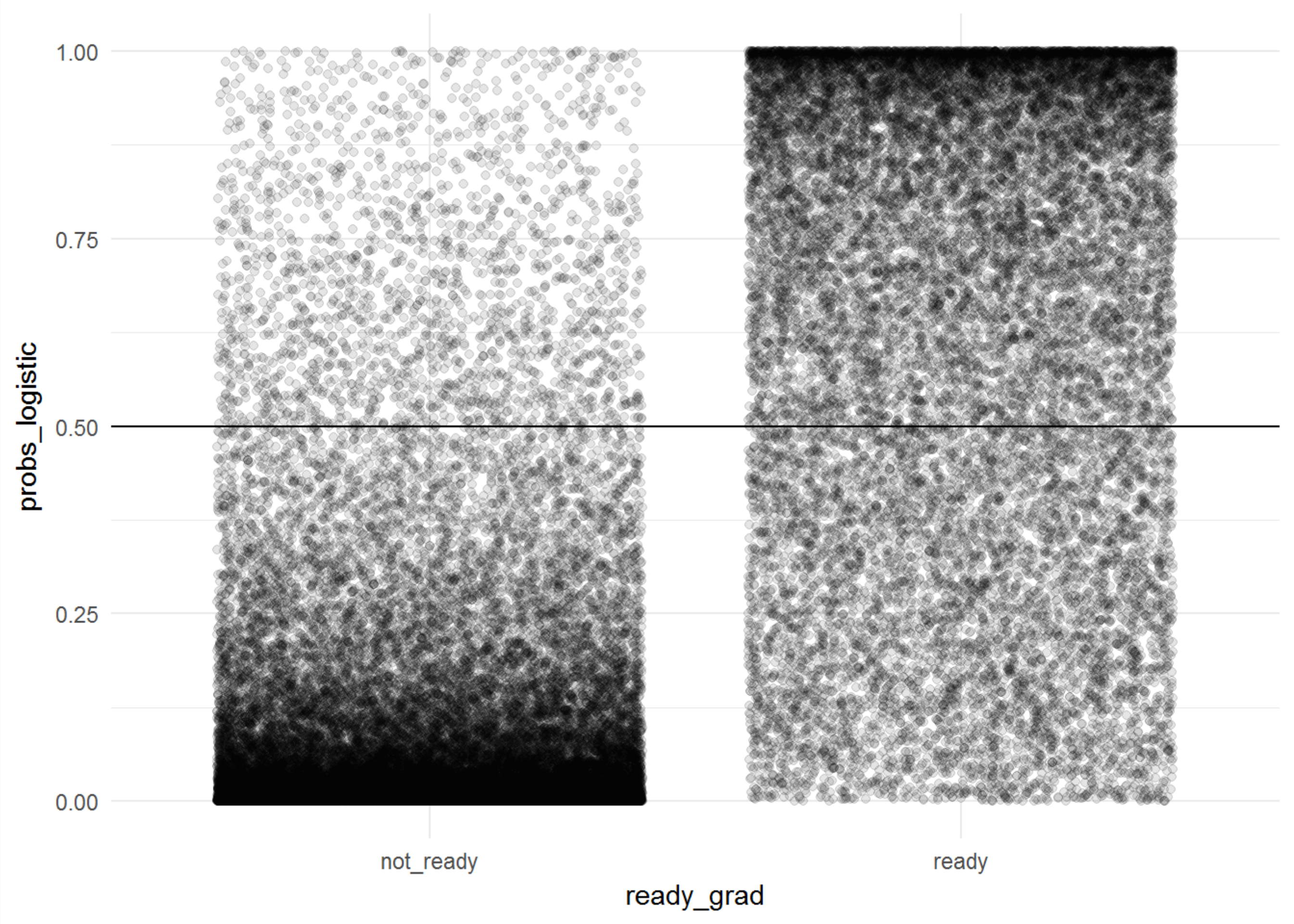
Use the fitted model to assign a probability that each student in the test set will be a ready graduate:

```
test_set$probs_logistic <- predict(model_logistic, test_x, type = "prob")$ready

ggplot(test_set, aes(x = ready_grad, y = probs_logistic)) + geom_jitter(alpha = 0.1) + theme_minimal()
```

Extracting Predictions from a Model

The model predicts that any student with `p > 0.5` is a ready graduate:



Extracting Predictions from a Model

Use the fitted model to predict on the test set and create a confusion matrix:

```
preds_logistic <- predict(model_logistic, test_x)
confusionMatrix(preds_logistic, test_y)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction  not_ready ready
##   not_ready     34193  6974
##   ready          2215 14398
##
##                Accuracy : 0.841
##                  95% CI : (0.838, 0.8439)
##     No Information Rate : 0.6301
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6424
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9392
##             Specificity : 0.6737
##          Pos Pred Value : 0.8306
```

Extracting Predictions from a Model

What does it look like for an 8th grader to have a **90 percent** predicted probability of ready graduation?

predictor	value
n_absences	16
enrollments	1
E	0
I	1
R	0
S	0
assault	0
weapons	0
theft_vandalism	0
sexual_assault_harassment	0

predictor	value
drugs_alcohol	0.00
threat	0.00
school_rules	1.00
bullying	0.00
fighting	0.00
scale_score_mt	0.42
scale_score_rd	2.18
school_scale_score_mt	-0.04
school_scale_score_rd	0.35
school_chronic_abs	14.69

Extracting Predictions from a Model

What does it look like for an 8th grader to have a **50 percent** predicted probability of ready graduation?

predictor	value
n_absences	18
enrollments	1
E	0
I	0
R	0
S	0
assault	0
weapons	0
theft_vandalism	0
sexual_assault_harassment	0

predictor	value
drugs_alcohol	0.00
threat	0.00
school_rules	0.00
bullying	0.00
fighting	0.00
scale_score_mt	0.85
scale_score_rd	0.89
school_scale_score_mt	0.58
school_scale_score_rd	0.51
school_chronic_abs	10.34

Extracting Predictions from a Model

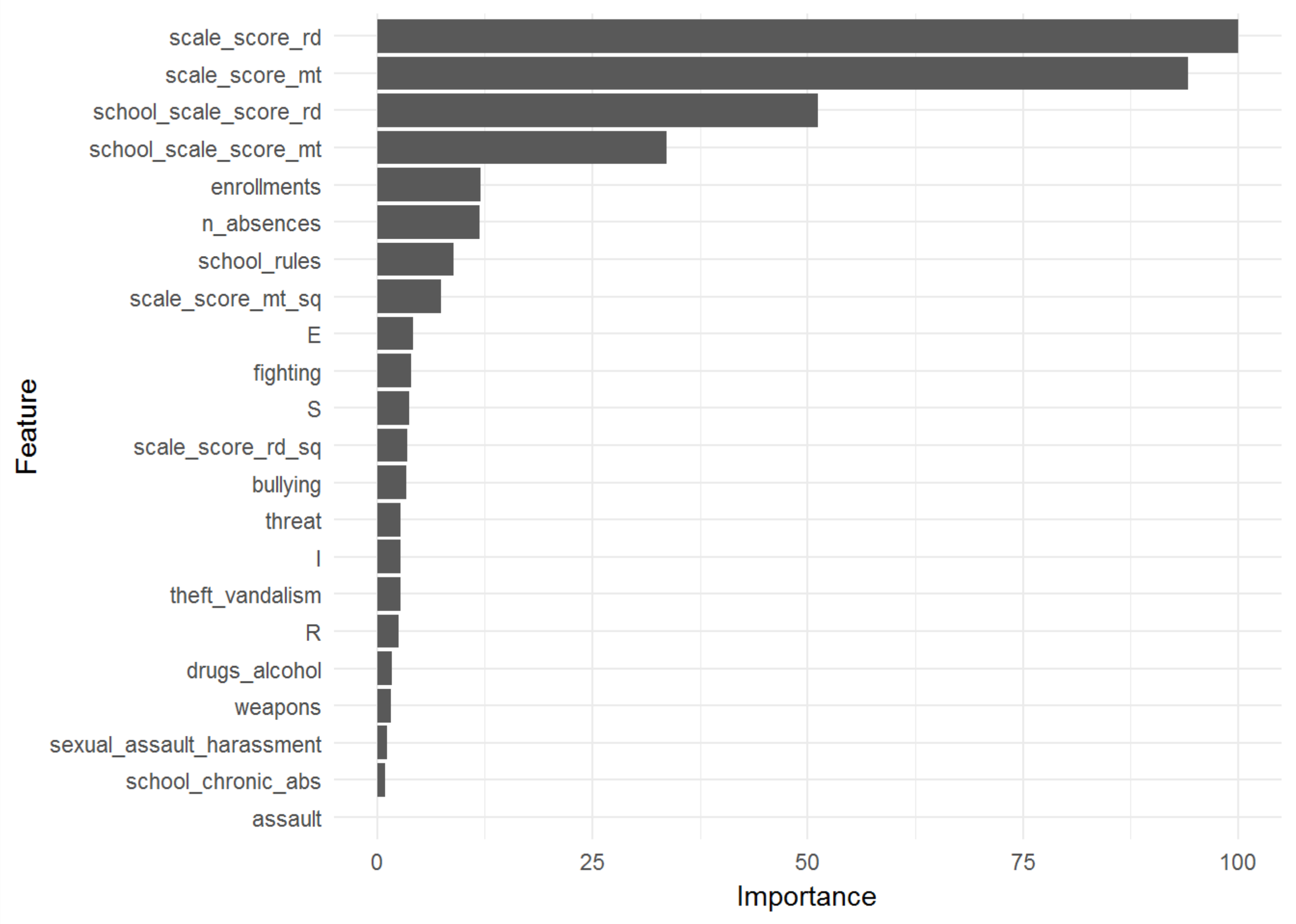
What does it look like for an 8th grader to have a **10 percent** predicted probability of ready graduation?

predictor	value
n_absences	8
enrollments	1
E	0
I	0
R	0
S	0
assault	0
weapons	0
theft_vandalism	0
sexual_assault_harassment	0

predictor	value
drugs_alcohol	0.00
threat	0.00
school_rules	0.00
bullying	0.00
fighting	0.00
scale_score_mt	-0.06
scale_score_rd	0.19
school_scale_score_mt	-0.56
school_scale_score_rd	-0.57
school_chronic_abs	22.69

Model Example

We can look at the relative importance of the predictors in predicting ready graduation:



Other Models

Repeat with other models:

- `gbm` (gradient boosting machine)
- `rpart` (recursive partitioning/decision tree)
- `rlda` (regularized linear discriminant analysis)
- `nnet` (neural network)
- `xgbLinear` (extreme gradient boosting w/ logistic regression)
- `xgbTree` (extreme gradient boosting w/ trees)

Just replace `method` in the call to `train`:

```
train(x = train_x, y = train_y, method = "", ...)
```

Other Models

Model descriptions (Applied Predictive Modeling, Kuhn & Johnson (2013)):

- `gbm` (gradient boosting machine)

The basic principle of boosting is to identify a "weak learner" (e.g. a CART tree) and a loss function (e.g. Accuracy or Kappa) the algorithm identifies an additive model that minimizes the loss function. The gradient, in GBM, refers to the residual error from the original model. After the first fit, the residual error is calculated and new learner is fit to the residuals - each model is subsequently added together for a user-specified number of iterations.

Other Models

Model descriptions (Applied Predictive Modeling, Kuhn & Johnson (2013)):

- `rpart` (recursive partitioning/decision tree)

Begin with the entire dataset, search every distinct value of every predictor and find the predictor and split value that maximizes the performance function (classification accuracy). Do this recursively for each predictor in the dataset, recursively partitioning the data by increasingly smaller differences in outcomes between the splits.

Other Models

Model descriptions (Applied Predictive Modeling, Kuhn & Johnson (2013)):

- `rlda` (regularized linear discriminant analysis)

Identify the linear combination of the predictors such that the between group variances was maximized relative to the within group variance. As the number of predictors grows, this deviates from the logistic regression solution because it is more flexible and employs more parameters - the additional parameters allow LDA to handle correlated predictors better than logistic regression.

Other Models

Model descriptions (Applied Predictive Modeling, Kuhn & Johnson (2013)):

- `nnet` (neural network)

Similar to a nonlinear regression model because linear regressors are combined into a set number of latent or hidden variables via a nonlinear function, and the values of the hidden units are summed to generate a linear prediction of the outcome.

Other Models

Model descriptions:

- `xgbTree` (extreme gradient boosting w/ trees)

Similar to GBM but includes column sub-sampling.

Other Models

Model descriptions:

- `xgbLinear` (extreme gradient boosting w/ logistic regression)

Similar to xgbLinear but uses a logistic regression.

Other Models

We end up with one prediction for each student for each model:

n_absences	enrollments	scale_score_mt	...	pred_gbm	pred_rpart	pred_rlda	pred_nnet	pred_xgblinear	pred_xgbtree
-0.0846197	-0.3042562	1.1150940	...	ready	ready	ready	ready	ready	ready
-0.2125824	-0.3042562	1.4633817	...	ready	ready	ready	ready	ready	ready
0.4272310	-0.3042562	-0.8308366	...	not_ready	not_ready	not_ready	not_ready	not_ready	not_ready
0.1713057	-0.3042562	-0.5609285	...	not_ready	not_ready	not_ready	not_ready	not_ready	not_ready
-0.4685077	-0.3042562	0.4107404	...	ready	ready	ready	ready	ready	ready
-0.9803584	-0.3042562	-0.8848182	...	not_ready	not_ready	not_ready	not_ready	not_ready	not_ready
-0.3405450	-0.3042562	0.3297680	...	not_ready	not_ready	not_ready	not_ready	not_ready	not_ready
-0.9803584	-0.3042562	1.7062990	...	ready	ready	ready	ready	ready	ready
-0.4685077	-0.3042562	0.8156025	...	ready	ready	ready	ready	not_ready	ready
-0.2125824	-0.3042562	0.8156025	...	ready	ready	ready	ready	ready	ready

Prediction Agreement

For the 57780 student in the test set, the count of models predicting that each student will be a ready graduate:

Models Predicting Ready	n
0	33445
1	2431
2	1602
3	1205
4	1065
5	1784
6	16248

Evaluating Predictions

Evaluating Predictions

Accuracy of all models statewide:

accuracy_gbm	accuracy_rpart	accuracy_rlda	accuracy_nnet	accuracy_xgblinear	accuracy_xgbtree
85.5	84.6	85.3	85.4	85.4	85.5

Evaluating Predictions

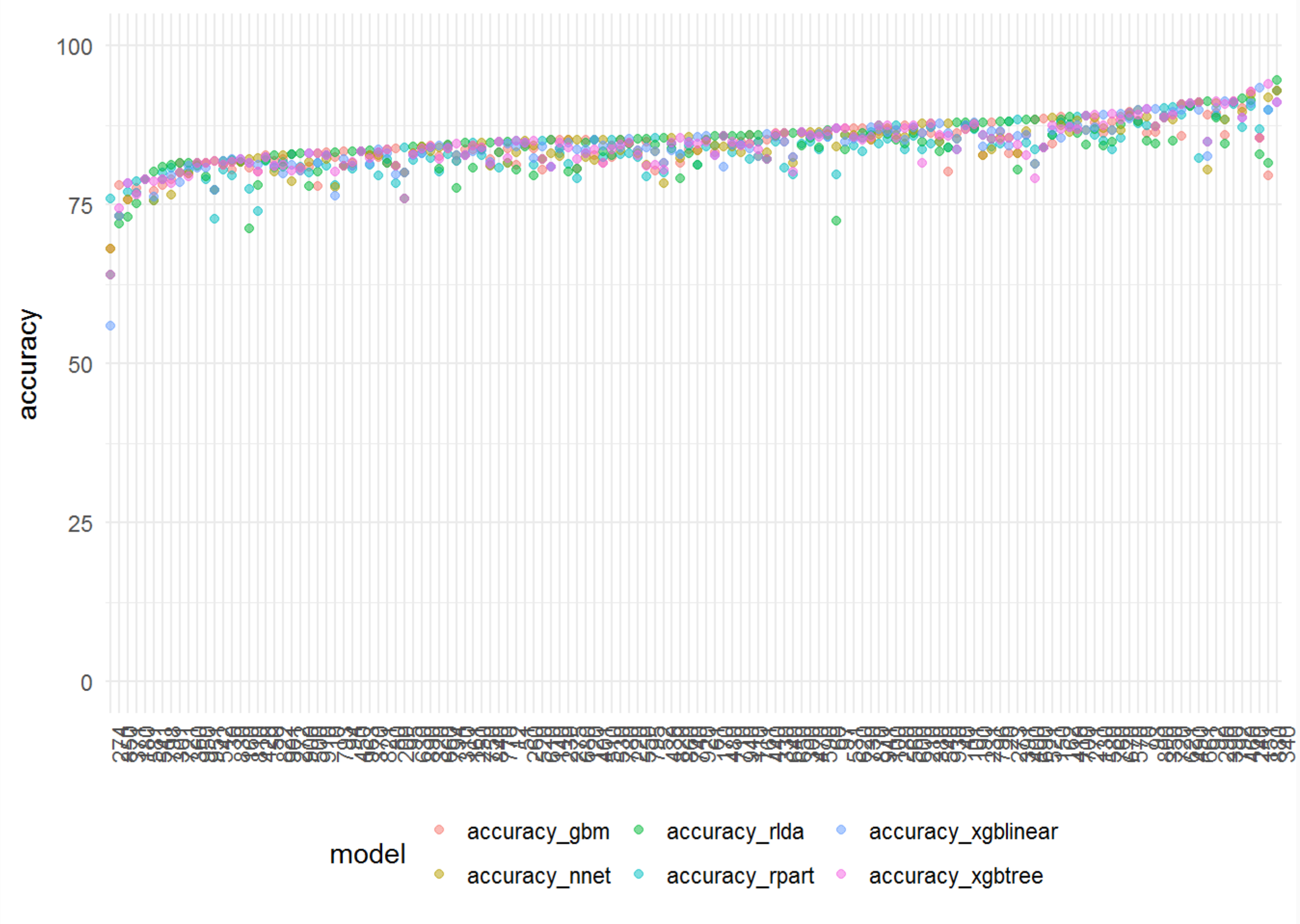
Accuracy of all models statewide:

accuracy_gbm	accuracy_rpart	accuracy_rlda	accuracy_nnet	accuracy_xgblinear	accuracy_xgbtree
85.5	84.6	85.3	85.4	85.4	85.5

Want to check that models perform well across districts, schools, student groups, etc.

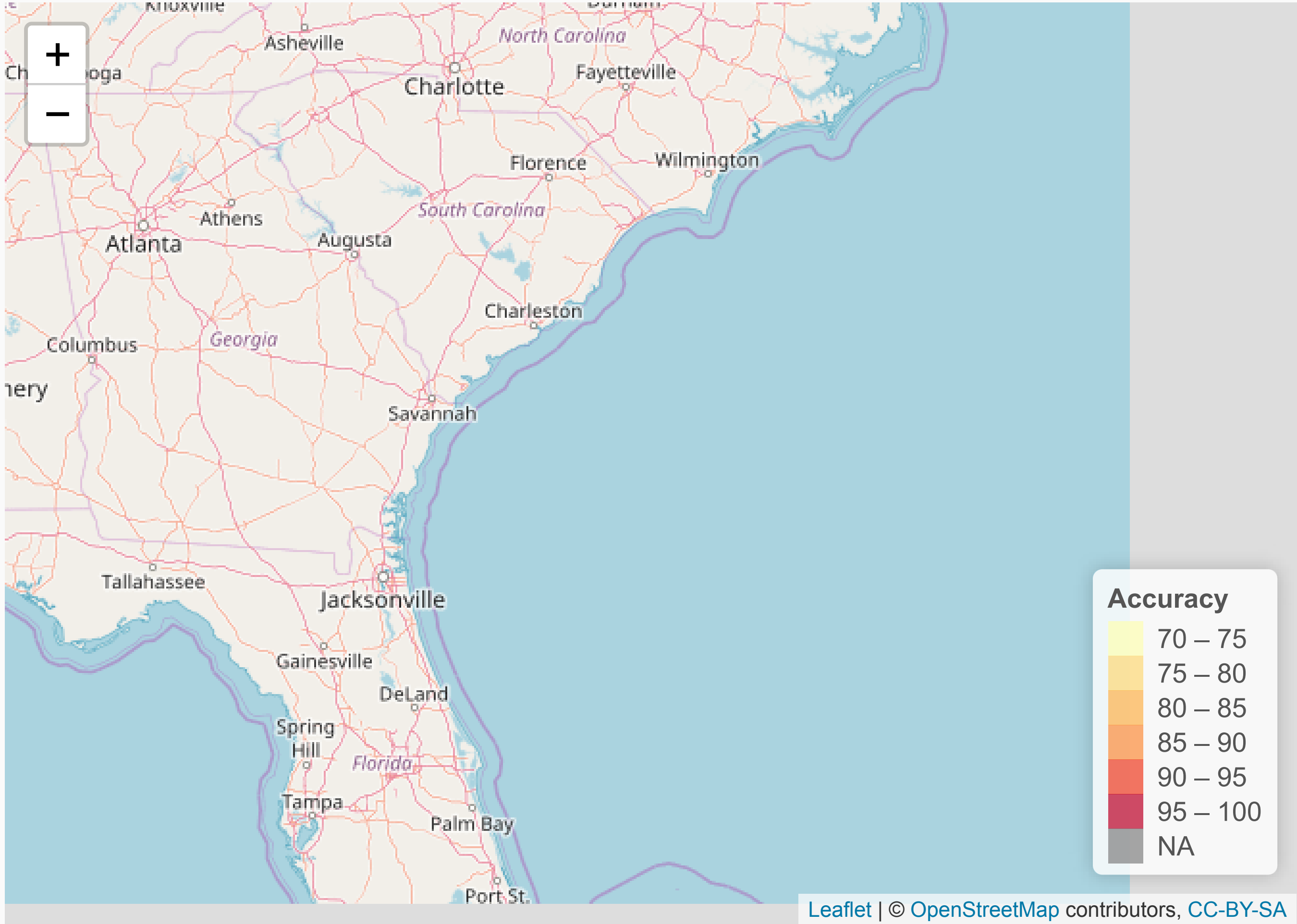
Evaluating Predictions

Accuracy of all models by district:



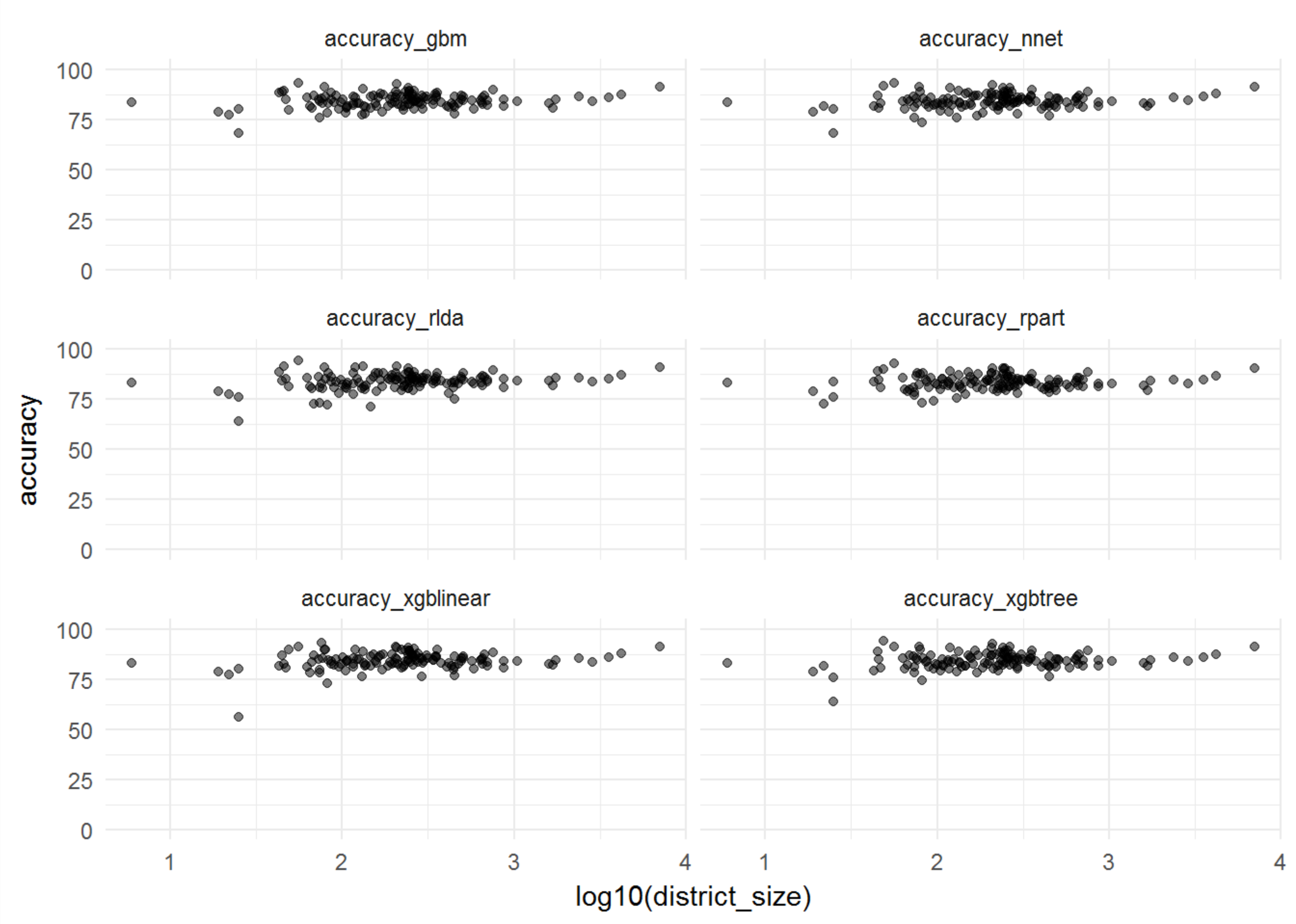
Evaluating Predictions

Accuracy of best model by district:



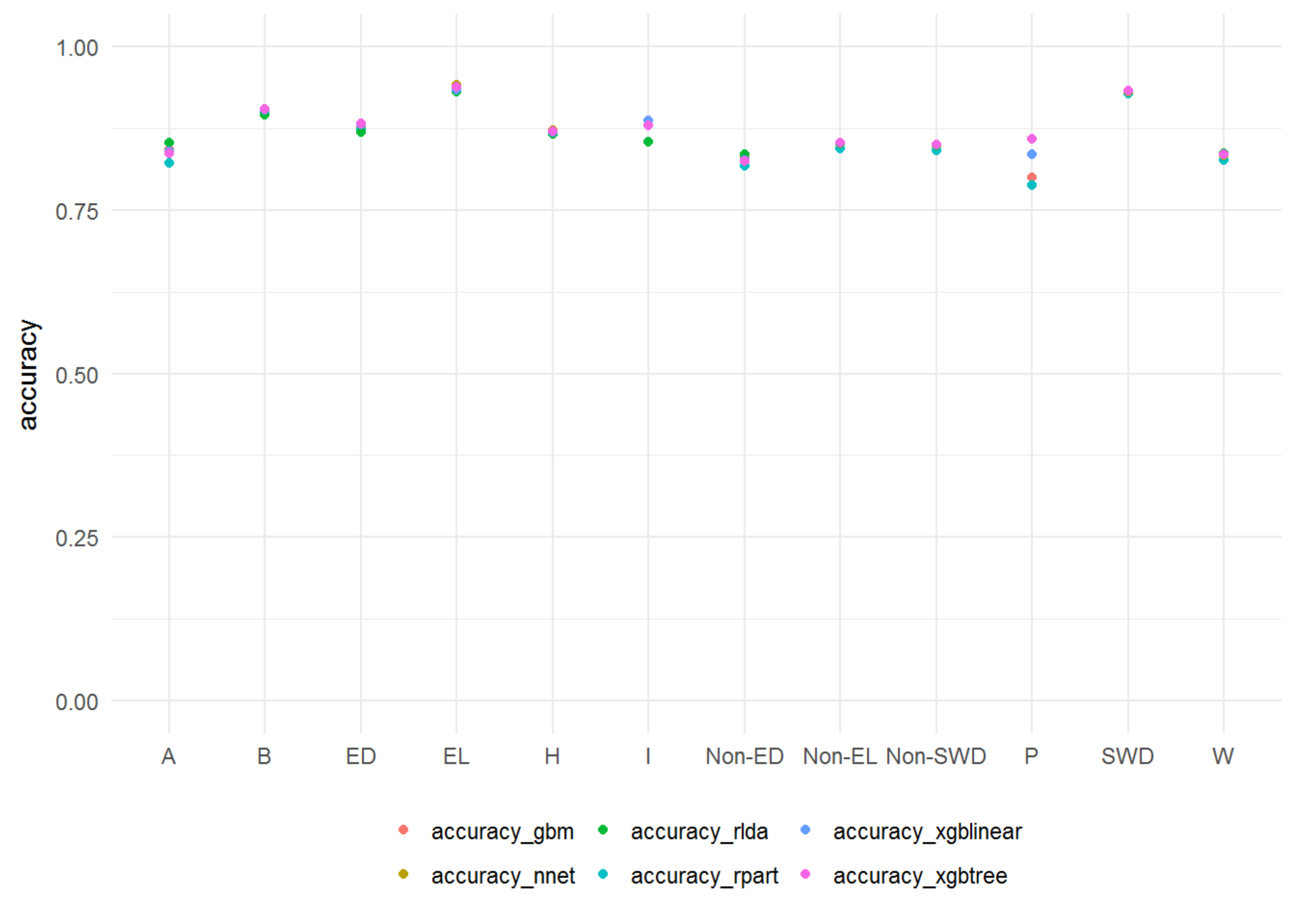
Evaluating Predictions

Accuracy by district size:



Evaluating Predictions

Accuracy by student group:



Evaluating Predictions

Recall from our logistic regression model, we assigned a probability and predicted that students with `p > 0.5` would be a ready graduate:

Evaluating Predictions

Beyond accuracy, two metrics of interest are **sensitivity** and **specificity**.

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  not_ready ready
##   not_ready     34193  6974
##   ready          2215 14398
##
##                Accuracy : 0.841
##                  95% CI : (0.838, 0.8439)
##     No Information Rate : 0.6301
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6424
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9392
##             Specificity : 0.6737
##          Pos Pred Value : 0.8306
##          Neg Pred Value : 0.8667
##              Prevalence : 0.6301
```

Sensitivity: Among predicted positives, how many are true positives?

$$Sensitivity = \frac{TruePositives}{TruePositives + FalseNegatives}$$

Evaluating Predictions

Beyond accuracy, two metrics of interest are **sensitivity** and **specificity**.

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction  not_ready ready
##   not_ready     34193  6974
##   ready          2215 14398
##
##                Accuracy : 0.841
##                  95% CI : (0.838, 0.8439)
##     No Information Rate : 0.6301
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6424
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9392
##             Specificity : 0.6737
##          Pos Pred Value : 0.8306
##          Neg Pred Value : 0.8667
##              Prevalence : 0.6301
```

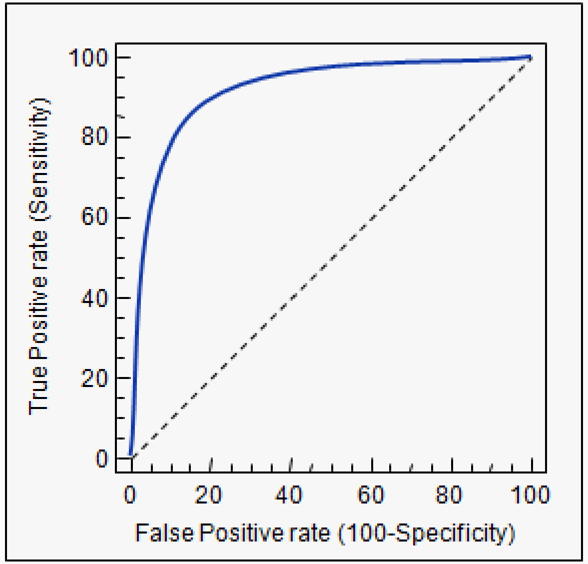
Specificity: Among predicted negatives, how many are true negatives?

$$Specificity = \frac{TrueNegatives}{TrueNegatives + FalsePositives}$$

Evaluating Predictions

If we plot $1 - specificity$ against $sensitivity$ for all thresholds of `p`, we get a **Receiver Operating Characteristic** curve. The **Area Under the Curve** or **AUC** is another metric of interest in evaluating Machine Learning predictions.

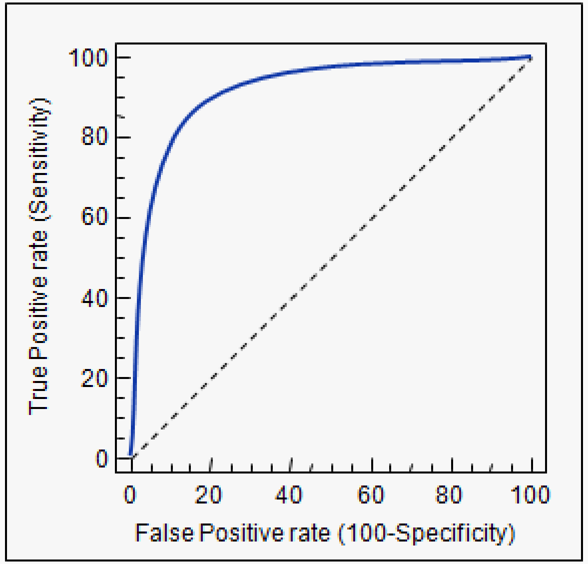
From StackExchange:



Evaluating Predictions

If we plot $1 - specificity$ against $sensitivity$ for all thresholds of `p`, we get a **Receiver Operating Characteristic** curve. The **Area Under the Curve** or **AUC** is another metric of interest in evaluating Machine Learning predictions.

From StackExchange:



Higher values of AUC indicate better prediction, where an AUC of 1 indicates perfect prediction (all ready grads predicted at 100% probability and all non-ready grads predicted at 0% probability).

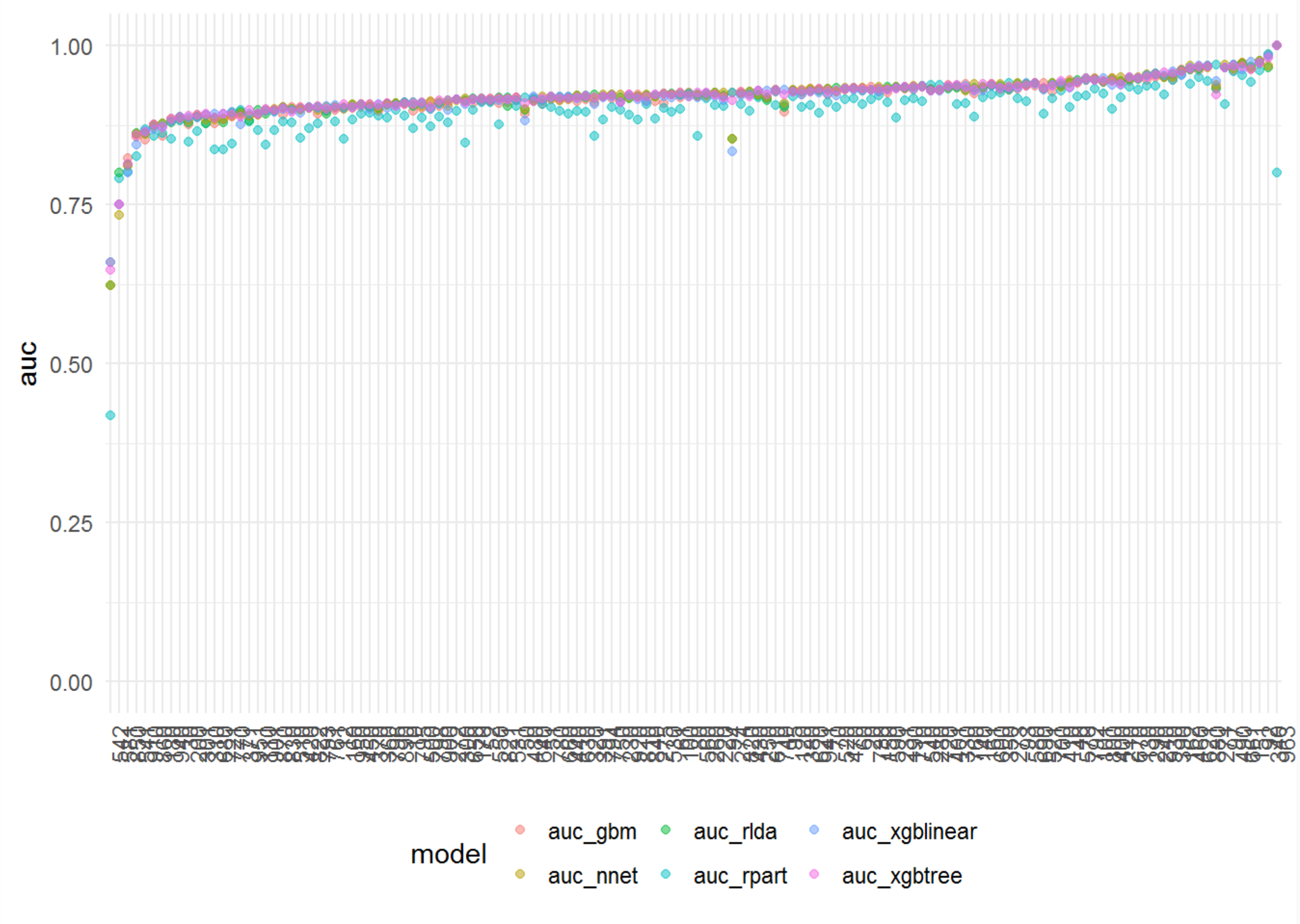
Evaluating Predictions

AUC for state:

auc_gbm	auc_rpart	auc_rlda	auc_nnet	auc_xgblinear	auc_xgbtree
0.9290424	0.9101992	0.9304181	0.9316807	0.9305663	0.9320431

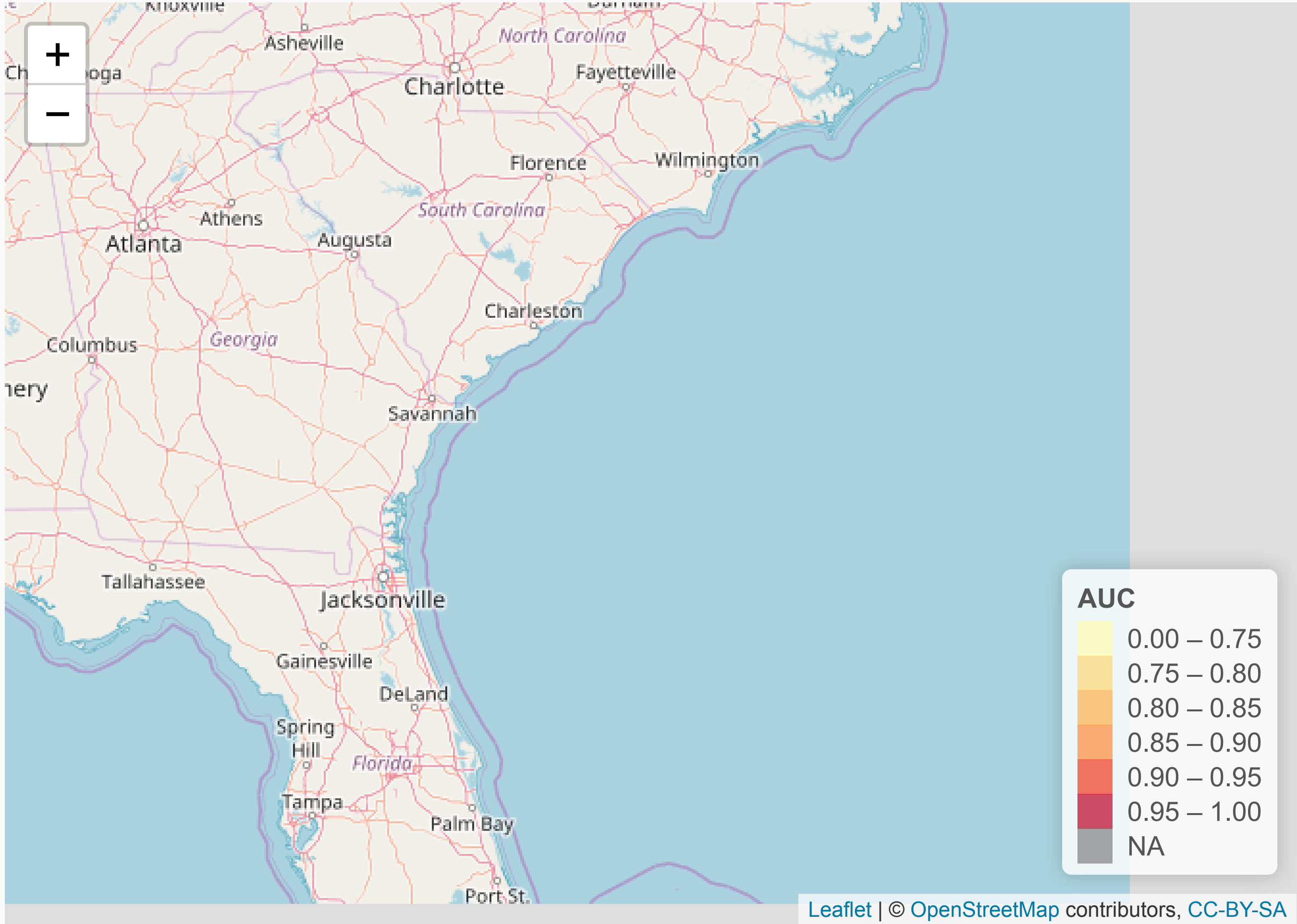
Evaluating Predictions

AUC by district:



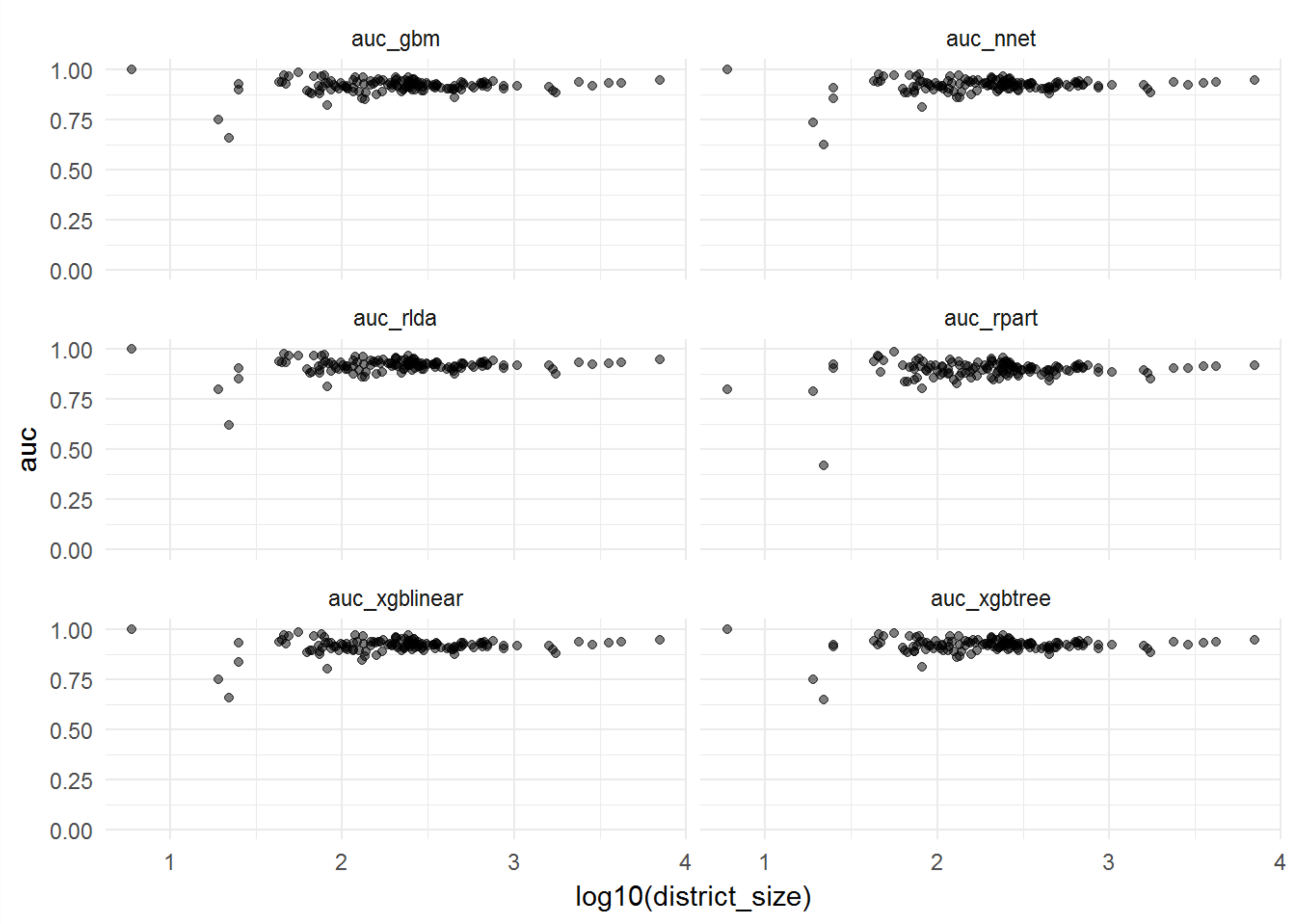
Evaluating Predictions

AUC of best model by district:



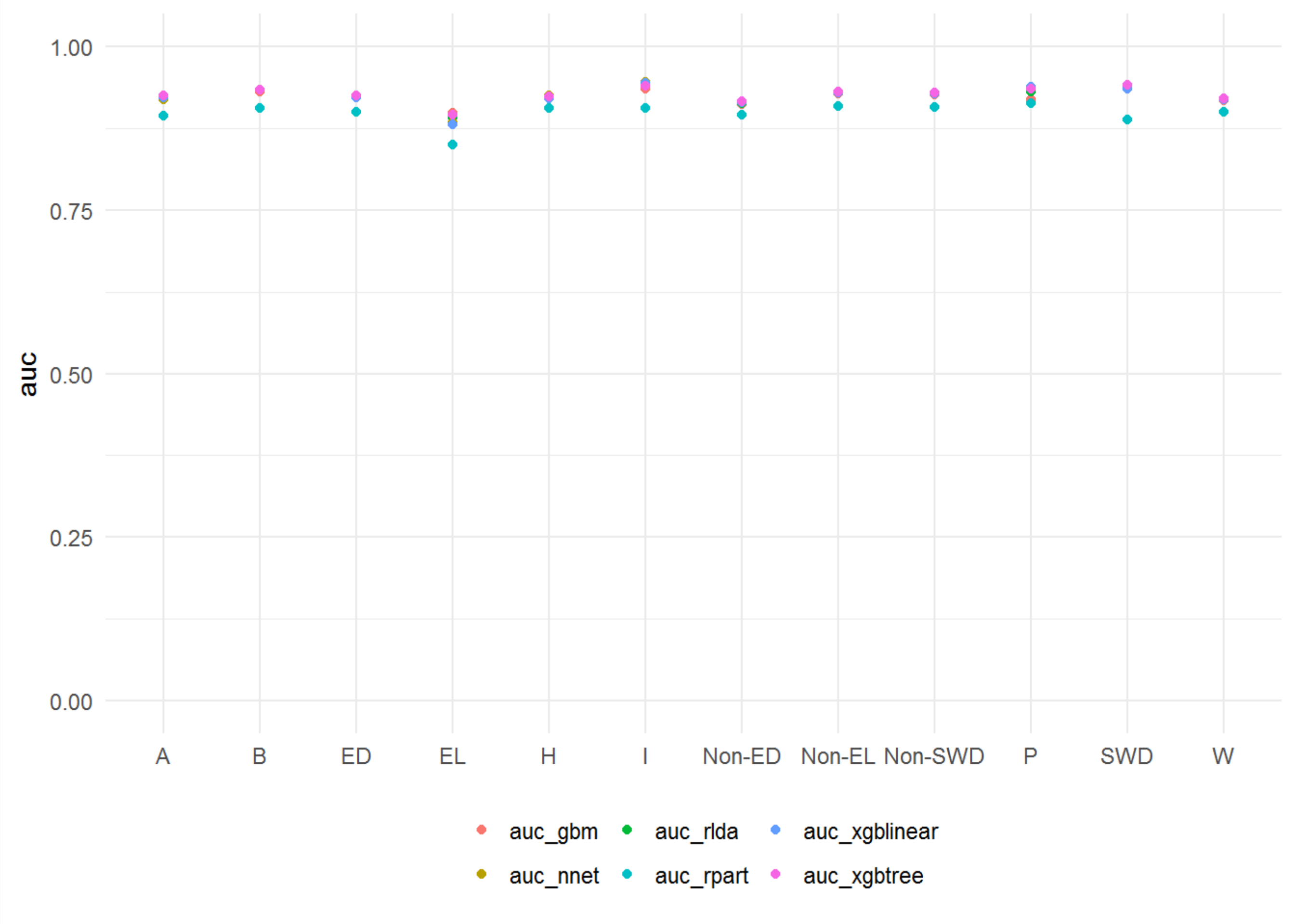
Evaluating Predictions

AUC by district size:



Evaluating Predictions

AUC by student group:



Evaluating Predictions

We can **ensemble** individual model predictions into a single prediction for each student by building a new model with ready graduation as the outcome and the probabilities from each model as the predictors:

prob_gbm	prob_rpart	prob_rlda	prob_nnet	prob_xgblinear	prob_xgbtree	ready_grad
0.0092406	0.0240583	0.0012994	0.0054117	0.0033301	0.0047055	not_ready
0.2368644	0.0240583	0.1265243	0.0652310	0.0990218	0.0794510	not_ready
0.0367904	0.0240583	0.0197076	0.0156224	0.0126924	0.0229484	not_ready
0.0055654	0.0240583	0.0022302	0.0055994	0.0022456	0.0038720	not_ready
0.0199687	0.0240583	0.0158360	0.0134914	0.0250610	0.0198719	not_ready
0.8953967	0.9154930	0.9929017	0.8550209	0.8638288	0.8902807	ready
0.0128640	0.1412815	0.0083384	0.0138672	0.0692797	0.0347961	not_ready
0.5872644	0.8289474	0.8729221	0.6328344	0.6229850	0.6696702	not_ready
0.0385679	0.0983051	0.0274910	0.0372662	0.0604229	0.0317129	not_ready

Recap

1. Model ready graduation with student test scores, absenteeism, mobility, discipline, school characteristics
2. Check for reasonable prediction performance across districts, schools, student groups
3. Ensemble multiple model predictions into one prediction
    - If ensemble results in improvement in performance, use ensembled model; otherwise can just use a single model

How do we use this?

How do we use this?

As an early warning system:

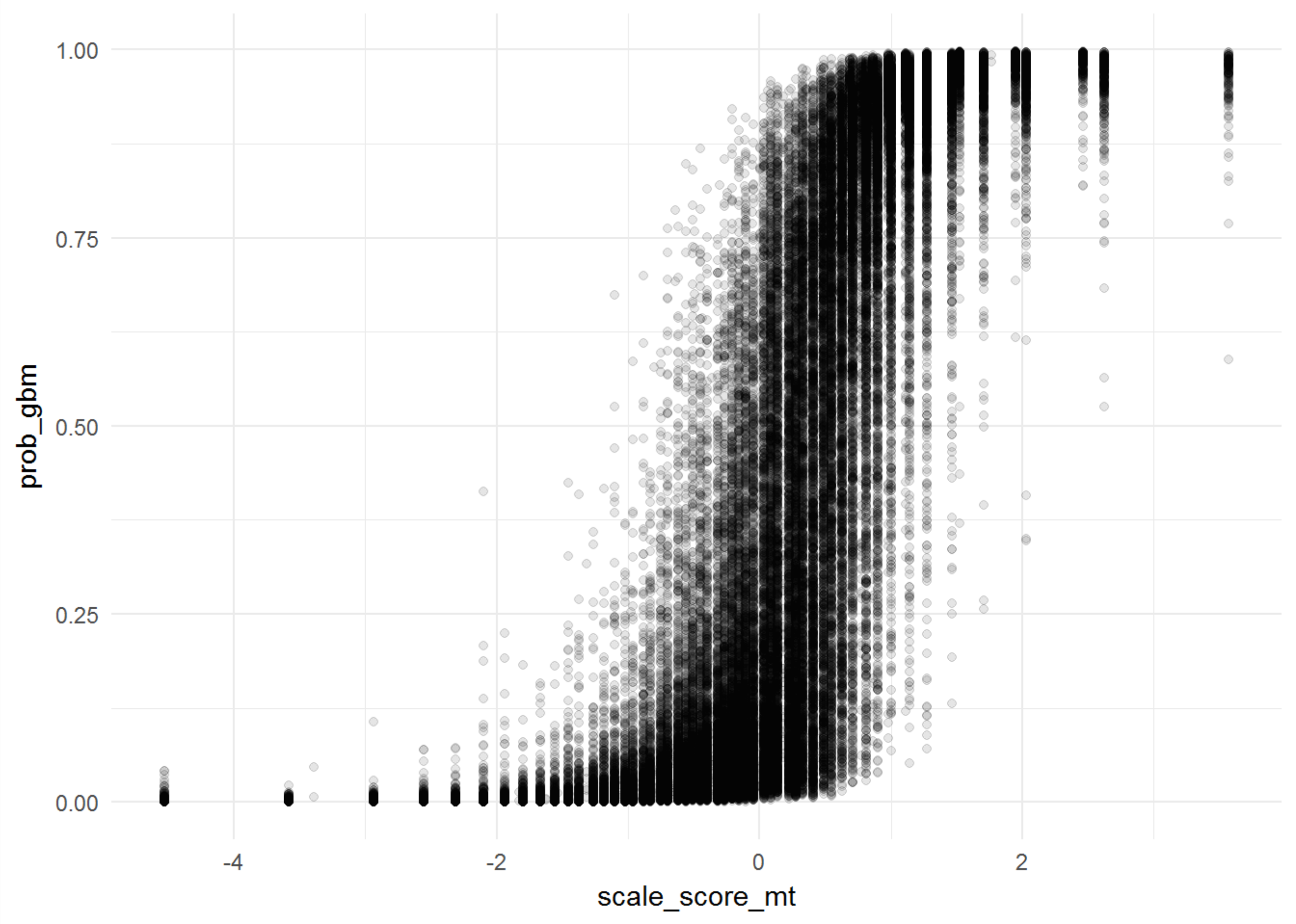
Send information to schools about students predicted not to be ready graduates, and which indicators are leading them to be predicted as not ready.

From Wisconsin DPI:

Name	Student ID	Status	Latest DEWS Outcome	DEWS Mobility	DEWS Discipline	DEWS Attendance	DEWS Assessments
--masked--	--masked--	Active	Moderate	Low	Low	High	High
--masked--	--masked--	Active	High	Low	Low	High	Low
--masked--	--masked--	Active	Low	Low	Low	High	Low
--masked--	--masked--	Active	Low	Low	Low	High	Low
--masked--	--masked--	Active	Low	Low	Low	High	Low
--masked--	--masked--	Active	Low	Low	Low	High	Moderate

How do we use this?

We can use the data to decide on thresholds for low, moderate, high risk:



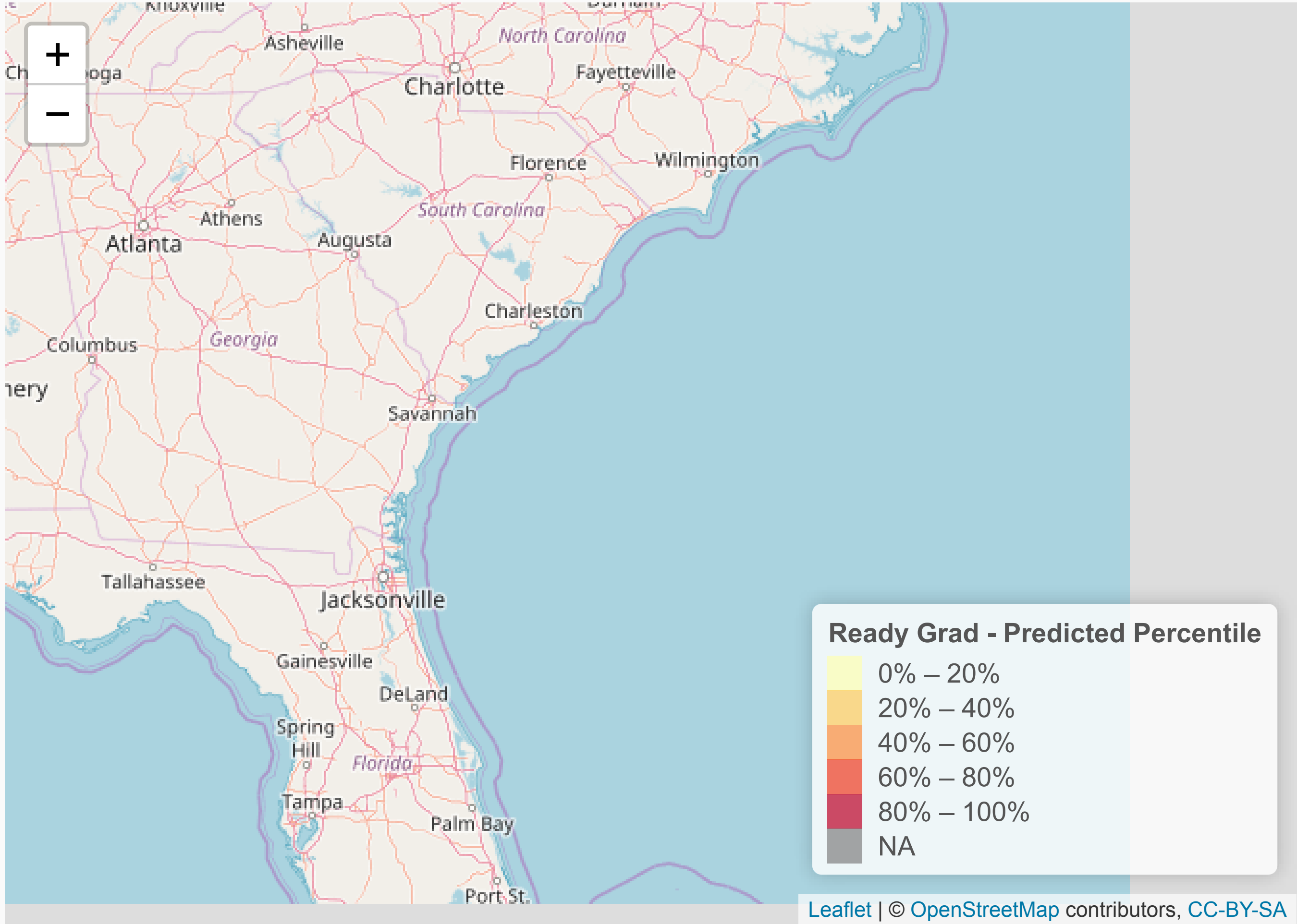
How do we use this?

As value-added by comparing actual % ready graduates to predicted % ready graduates:

system	ready_grad	pred_gbm	diff
10	0.3668763	0.3836478	-0.0167715
12	0.5584906	0.6150943	-0.0566038
20	0.2913386	0.2145669	0.0767717
30	0.3405797	0.3115942	0.0289855
40	0.2844828	0.2500000	0.0344828
50	0.3803030	0.3469697	0.0333333
51	0.4953271	0.5140187	-0.0186916
52	0.6258503	0.5170068	0.1088435
60	0.3755725	0.3160305	0.0595420
61	0.4095238	0.3936508	0.0158730
70	0.1904762	0.2016807	-0.0112045

How do we use this?

As value-added by comparing actual % ready graduates to predicted % ready graduates:



Recap

1. Model ready graduation with student test scores, absenteeism, mobility, discipline, school characteristics
2. Check for reasonable prediction performance across districts, schools, student groups
3. Ensemble multiple model predictions into one prediction
   - If ensemble results in improvement in performance, use ensembled model; otherwise can just use a single model
4. Deploy as Early Warning System or Value-Added or other use?

The End

Code and presentation: https://github.com/tnedu/grad-prediction