Predicting Purchases, Rare Diseases, and More: Using Ordinal Regression to Estimate Rare Event Probabilities

Get these slides, code, and more

Greg Faletto

Data Scientist at VideoAmp

Ph.D. in Statistics, University of Southern California Marshall School of Business

Email: gregory.faletto@marshall.usc.edu

Twitter: @gregoryfaletto

August 12th, 2023

Data Con LA 2023

Outline

1 Background and Practical Motivation

2 The Proportional Odds Model

3 Training the Proportional Odds Model

4 PRESTO

5 Data Application

Get these slides and code at https://github.com/gregfaletto/dataconla2023 (or scan the QR code)

Estimating Probabilities of Rare Events is Important and Hard

Classifiers struggle in the *class imbalance* setting where one class is very rare.

Unfortunately, accurate estimates of the probabilities of rare events are often very important.

> **Online marketing:** clicking on ads and making a purchase is very rare.
>
> However, it’s important to accurately estimate the probability of a user making a purchase, because click ads are sold by auction.
>
> Ad buyers need to know the probability a user will make a purchase as accurately as possible in order to bid effectively.

Estimating Probabilities of Rare Events is Important and Hard

Classifiers struggle in the *class imbalance* setting where one class is very rare.

Unfortunately, accurate estimates of the probabilities of rare events are often very important.

> **Health and medicine:** rare diseases can be expensive to treat, the resources to deliver effective treatment are scarce, and it can be very stressful to believe you may have a rare disease. Accurately estimating the probability of having a rare disease is crucial to treat patients effectively.

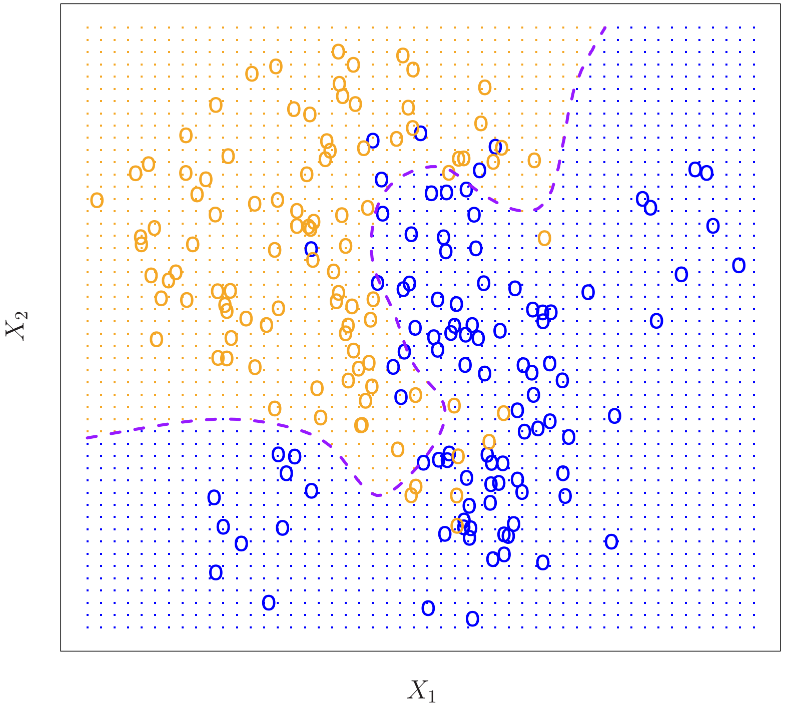
Bayes Decision Boundary

Setting: binary classification.

In the **orange region**, observations have a higher probability of being in the **orange class**. In the **purple region**, observations have a higher probability of being in the **purple class**.

The **Bayes decision boundary** is the set of points right in the middle (the dashed line). Observations that lie exactly on this line have exactly 50% probability of being in each class.

(See Section 2.2 of *An Introduction to Statistical Learning with Applications in R:* James et. al 2021).



Source: Figure 2.13 from James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An Introduction to Statistical Learning with Applications in R*. Second edition. New York: Springer.

Bayes Decision Boundary

Example: logistic regression models learn a **linear** decision boundary between the classes.

The logistic regression model for the conditional probability that an observation with predictors $\boldsymbol{x} = (x_1, x_2, \ldots, x_p)$ lies in class 2, $\mathbb{P}(y = 2 \mid \boldsymbol{x})$, can be written as

$$\ln\left(\frac{\mathbb{P}\left(y = 1 \mid \boldsymbol{x}\right)}{\mathbb{P}\left(y = 2 \mid \boldsymbol{x}\right)}\right) = \alpha + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p,$$

where α is an intercept and $\beta_1, \beta_2, \ldots, \beta_p$ are coefficients.

If the model is correct, when we plug in the true coefficients, the right side of this equation defines the (linear) decision boundary.

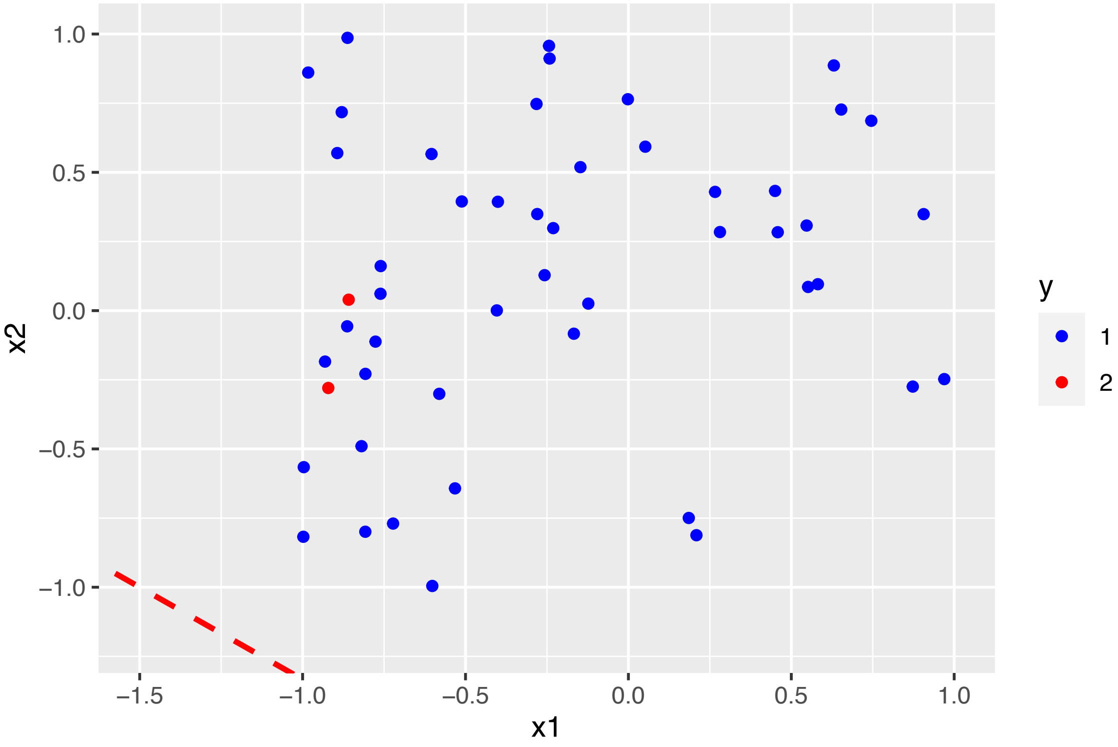
When we train a logistic regression model, the estimated coefficients make an equation for the estimated decision boundary.

Classification of Rare Events

For rare events, even if logistic regression is a good model, the data might look more like this than the previous slide.

I generated this data from a logistic regression model. Dashed line: Bayes decision boundary.

The data are all far away from the decision boundary. (That’s why the red outcome is rare—none of the observations have close to a 50% chance of falling in the red category.)

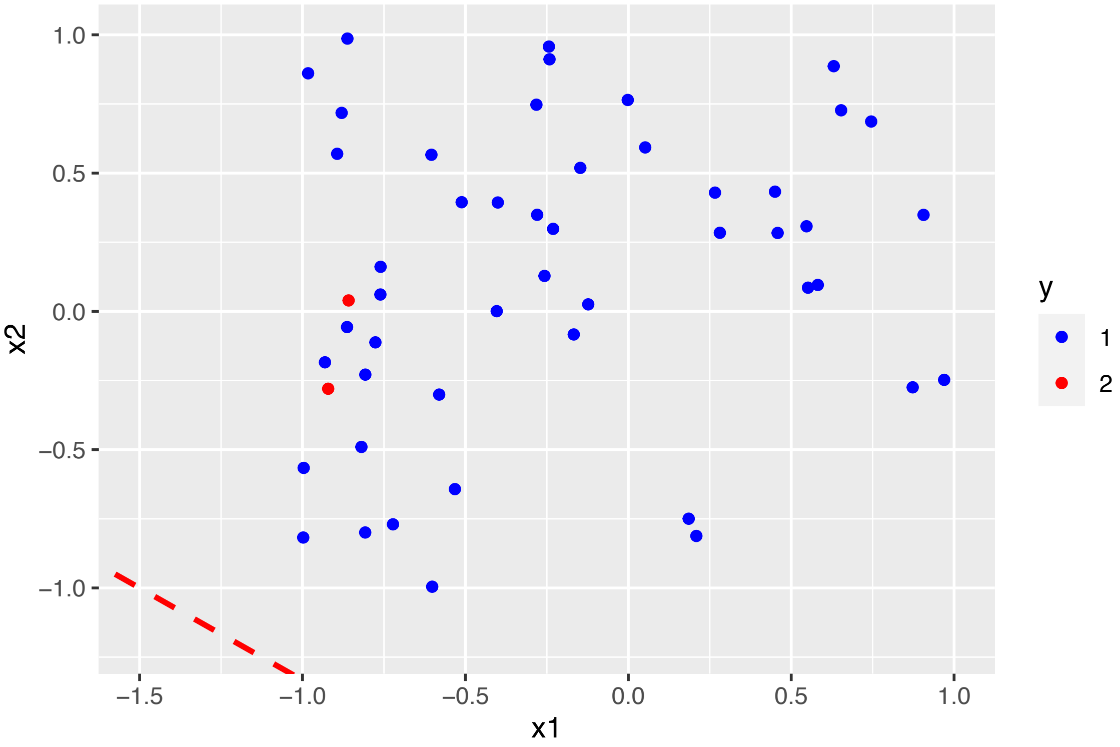


Classification of Rare Events

Because the red class is rare, we don't observe many points near the decision boundary.

It's easiest to estimate the decision boundary precisely when we observe points near it. In this case, estimating the decision boundary, and $\mathbb{P}(y = 2 \mid \boldsymbol{x})$, is **very difficult.**

In fact, I tried estimating a logistic regression model on this data…

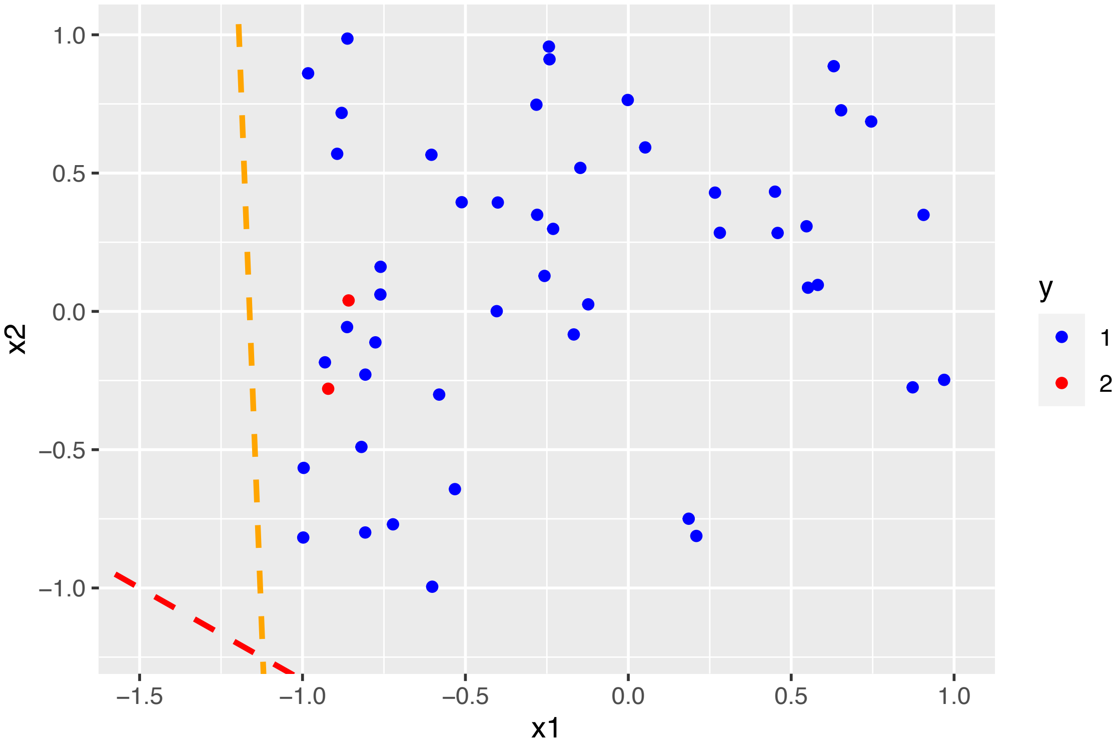


Classification of Rare Events

Because the red class is rare, we don't observe many points near the decision boundary.

It's easiest to estimate the decision boundary precisely when we observe points near it. In this case, estimating the decision boundary, and $\mathbb{P}(y = 2 \mid \boldsymbol{x})$, is **very difficult.**

In fact, I tried estimating a logistic regression model on this data…



The estimated decision boundary (orange) is very bad $\Longrightarrow$ very bad rare probability estimates. (Not much better even with very large data sets!)

A Possible Solution: Leveraging Data From More Common Outcomes

Sometimes there are more common intermediate (**ordered**) outcomes.

> **Online marketing:** clicking on ads without making a purchase is much more common than clicking on ads and making a purchase.
>
> **Health and medicine:** sometimes there are intermediate outcomes between having a disease and being completely healthy (normal blood glucose > prediabetes > diabetes, normal blood pressure > prehypertension > hypertension, etc.)

Estimating the decision boundaries between these more common outcomes is easier (less class imbalance).

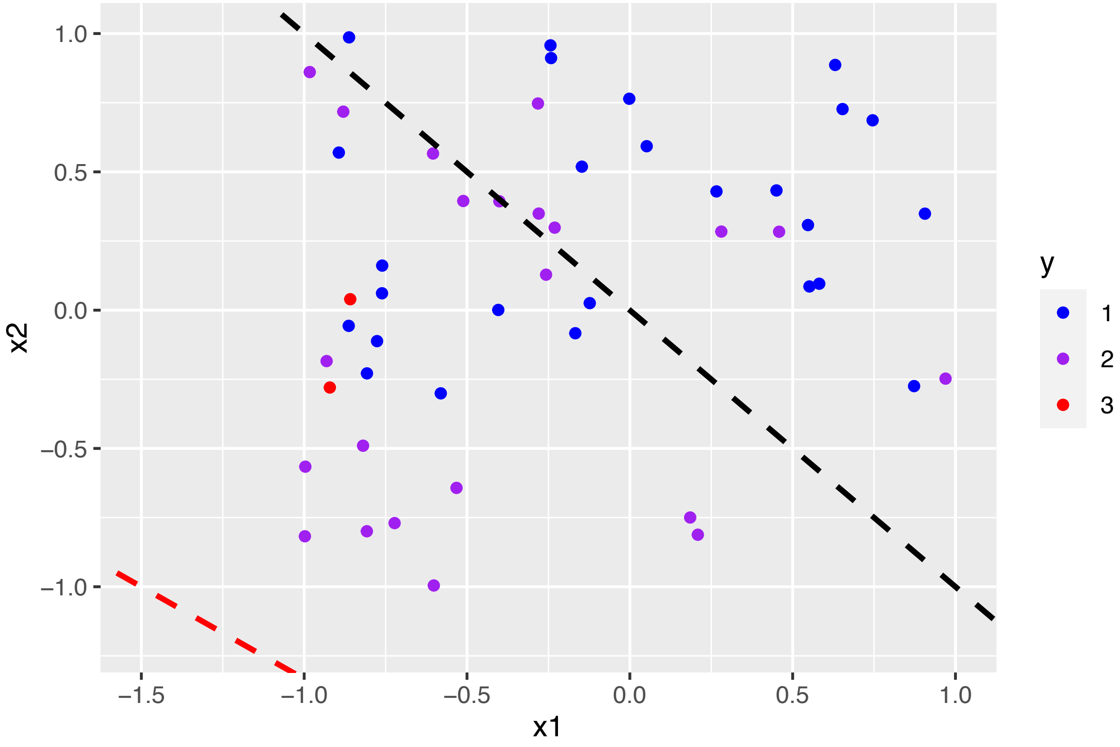
Idea: maybe the factors that are predictive of common outcomes can be used to improve predictions of rare outcomes.

A Possible Solution: Leveraging Data From More Common Outcomes

Same data as before, same red decision boundary as before. Class 2 is an intermediate outcome between class 1 and class 3. (In previous slide, classes 1 and 2 were combined.)

Black dashed line: Bayes decision boundary between classes 1 and 2.

Abundant data near decision boundary between classes 1 and 2 $\Longrightarrow$ easier to learn.



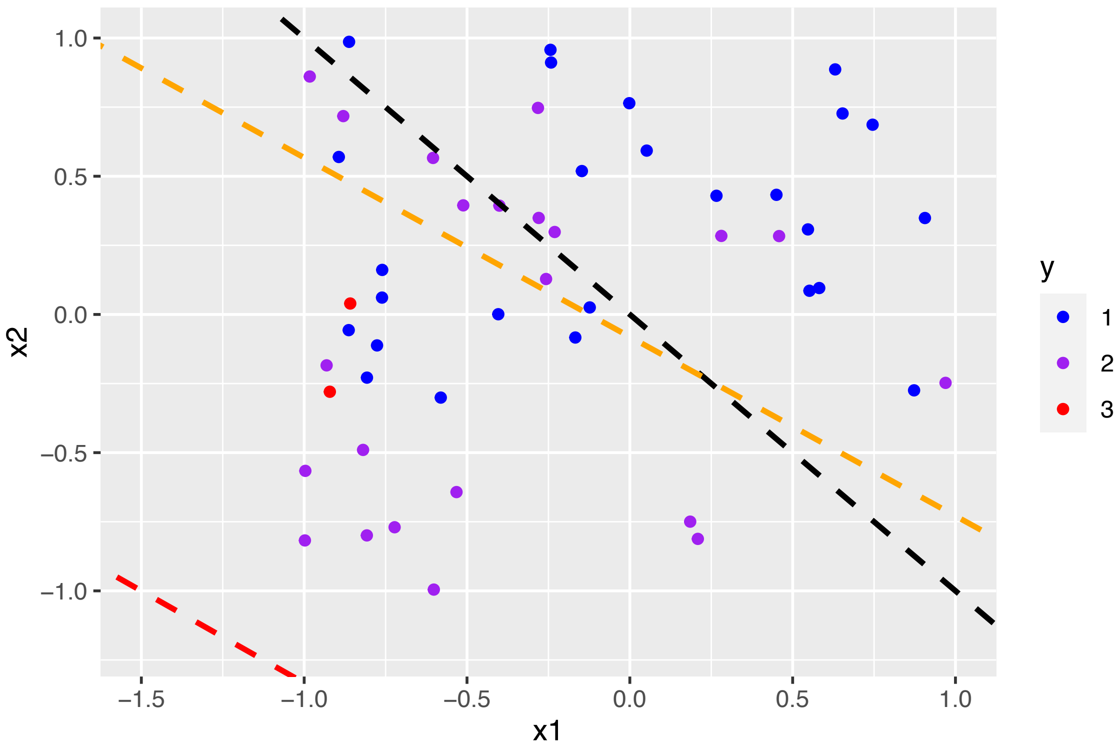
A Possible Solution: Leveraging Data From More Common Outcomes

Same data as before, same red decision boundary as before. Class 2 is an intermediate outcome between class 1 and class 3. (In previous slide, classes 1 and 2 were combined.)

Black dashed line: Bayes decision boundary between classes 1 and 2.

Abundant data near decision boundary between classes 1 and 2 $\Longrightarrow$ easier to learn.

The estimated decision boundary between classes 1 and 2 (orange) does seem to fit much better.



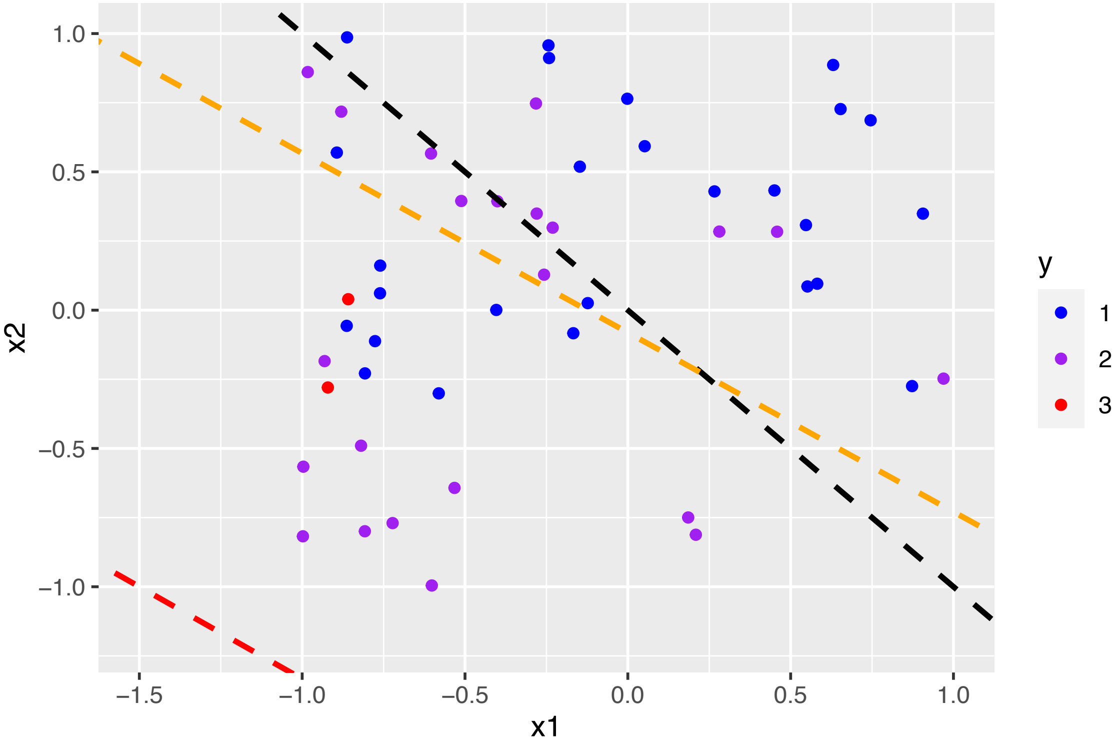
A Possible Solution: Leveraging Data From More Common Outcomes

Same data as before, same red decision boundary as before. Class 2 is an intermediate outcome between class 1 and class 3. (In previous slide, classes 1 and 2 were combined.)

Black dashed line: Bayes decision boundary between classes 1 and 2.

Abundant data near decision boundary between classes 1 and 2 $\Longrightarrow$ easier to learn.

The estimated decision boundary between classes 1 and 2 (orange) does seem to fit much better.



IDEA: Maybe we can leverage our more precise estimate of the decision boundary between classes 1 and 2 to better estimate the decision boundary between classes 2 and 3 $\Longrightarrow$ better estimated probabilities of lying in class 3.

Outline

1 Background and Practical Motivation

2 The Proportional Odds Model

3 Training the Proportional Odds Model

4 PRESTO

5 Data Application

Get these slides and code at https://github.com/gregfaletto/dataconla2023 (or scan the QR code)

The Proportional Odds Model

Remember: The logistic regression model for $\mathbb{P}(y = 2 \mid \boldsymbol{x})$ can be written as

$$\ln\left(\frac{\mathbb{P}\left(y = 1 \mid \boldsymbol{x}\right)}{\mathbb{P}\left(y = 2 \mid \boldsymbol{x}\right)}\right) = \alpha + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p,$$

where α is an intercept and $\beta_1, \beta_2, \ldots, \beta_p$ are coefficients.

The *proportional odds model* or *ordered logit model* (McCullagh, 1980) is a model for ordinal (ordered) outcomes $\{1,2,3,\ldots,K\}$. It models the decision boundary between consecutive categories using logistic regression models, but it assumes that the decision boundaries all have the same $\boldsymbol{\beta}$ vector and $K-1$ different intercepts:

$$\ln\left(\frac{\mathbb{P}\left(y \leq k \mid \boldsymbol{x}\right)}{\mathbb{P}\left(y > k \mid \boldsymbol{x}\right)}\right) = \alpha_k + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p, \qquad k = 1,\ldots,K-1.$$

The Proportional Odds Model

The *proportional odds model* or *ordered logit model* (McCullagh, 1980) is a model for ordinal (ordered) outcomes $\{1,2,3,\ldots,K\}$. It models the decision boundary between consecutive categories using logistic regression models, but it assumes that the decision boundaries all have the same $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_p)$ vector and $K-1$ different intercepts:

$$\ln\left(\frac{\mathbb{P}\left(y \leq k \mid \boldsymbol{x}\right)}{\mathbb{P}\left(y > k \mid \boldsymbol{x}\right)}\right) = \alpha_k + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p, \qquad k = 1,\ldots,K-1.$$

Basically, it **jointly estimates logistic regression models** for each of the $K-1$ decision boundaries between these ordered outcomes that borrow the same $\boldsymbol{\beta}$ parameters for all of the decision boundaries.

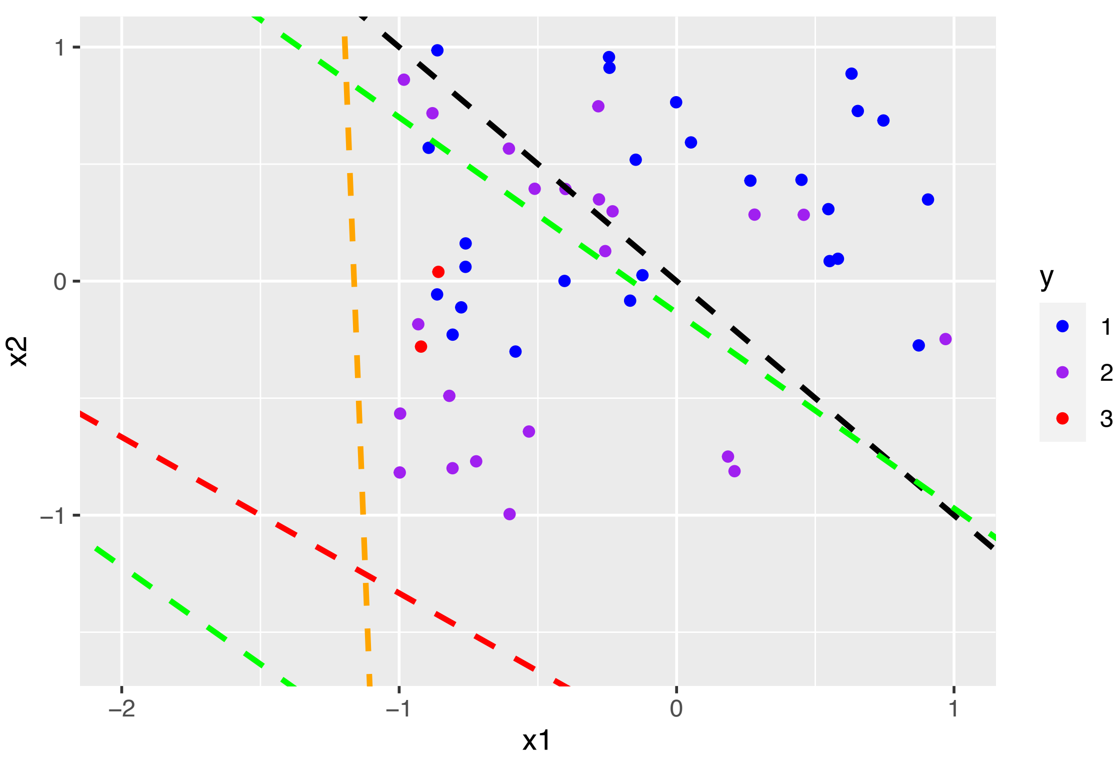
Only have to estimate one **unique** parameter for each of the $K-1$ decision boundaries: α_k. (Assumption of equal $\boldsymbol{\beta}$ means we don't have to learn much from scarce data in rare categories.)

The Proportional Odds Model Leverages Data From More Common Outcomes

Green dashed lines: estimated decision boundaries from proportional odds model. (Orange estimated decision boundary is from logistic regression on rare class, same as before.)

The proportional odds model does seem to yield a much better estimate of the rare decision boundary!

Also yields better estimates of $\mathbb{P}(y = 3 \mid \boldsymbol{x})$.



Outline

1 Background and Practical Motivation

2 The Proportional Odds Model

3 Training the Proportional Odds Model

4 PRESTO

5 Data Application

Get these slides and code at https://github.com/gregfaletto/dataconla2023 (or scan the QR code)

Training the Proportional Odds Model

The proportional odds model is implemented in R in the `MASS` package, in the function `polr`. In the next few slides, we'll walk through R code that trains a proportional odds model on an example data set.

In Python, the proportional odds model is implemented in the `OrderedModel` function in the `statsmodels` library. Documentation is available at https://www.statsmodels.org/stable/examples/notebooks/generated/ordinal_regression.html#Logit-ordinal-regression:, or you can scan the QR code below. (Go to the "Logit ordinal regression" section of the page.)

Data Set

As an example, we will use the `soup` data set from the `ordinal` R package. It is a set of responses in a taste-testing study from a soup company.

Each soup taster was given a reference soup. Then they were blindly given a sample of either the reference soup or a new test soup. The taster then said whether they thought the sample was the reference soup or a new test soup. They also said how confident they were in their response (“sure”, “not sure”, or “guess”).

This results in a 6-level ordered response variable, ranging from “reference, sure” to “not reference, sure.”

The data set contains 1847 responses, as well as 9 predictors we will use associated with the taster, the sample they were given, and the circumstances of the tasting.

Loading the Data Set

First we will install the `ordinal` package, load the `soup` data set, and take a quick look at it (on the next slide).

```
> install.packages(“ordinal”)
> library(ordinal)
> data(soup)
```

Examining the Data Set

```
> str(soup)
'data.frame': 1847 obs. of  12 variables:
 $ RESP    : Factor w/ 185 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1
1 ...
 $ PROD    : Factor w/ 2 levels "Ref","Test": 1 2 1 2 1 2 2 2 2 1 ...
 $ PRODID  : Factor w/ 6 levels "1","2","3","4",..: 1 2 1 3 1 6 2 4 5
1 ...
 $ SURENESS: Ord.factor w/ 6 levels "1"<"2"<"3"<"4"<..: 6 5 5 6 5 5 2 5
5 2 ...
 $ DAY     : Factor w/ 2 levels "1","2": 1 1 1 1 2 2 2 2 2 2 ...
 $ SOUPTYPE: Factor w/ 3 levels "Self-made","Canned",..: 2 2 2 2 2 2 2 2
2 2 ...
 $ SOUPFREQ: Factor w/ 3 levels ">1/week","1-4/month",..: 1 1 1 1 1 1 1
1 1 1 ...
 $ COLD    : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 2 ...
 $ EASY    : Factor w/ 10 levels "1","2","3","4",..: 7 7 7 7 7 7 7 7 7 7
...
 $ GENDER  : Factor w/ 2 levels "Male","Female": 2 2 2 2 2 2 2 2 2 2 ...
 $ AGEGROUP: Factor w/ 4 levels "18-30","31-40",..: 4 4 4 4 4 4 4 4 4
4 ...
 $ LOCATION: Factor w/ 3 levels "Region 1","Region 2",..: 1 1 1 1 1 1 1
1 1 1 ...
```

Examining the Data Set

We can examine the proportions of observations in each class to see that observations in class 1 (“reference, sure”) are relatively rare.

```
> summary(soup$SURENESS)/length(soup$SURENESS)
         1          2          3          4          5          6 
0.12344342 0.14076881 0.06226313 0.05305901 0.14997293 0.47049269 
```

Encoding the Response as an Ordinal Variable

The `polr` function requires the response to be encoded as an ordered categorical variable. Our response, `SURENESS`, is already encoded as an ordinal variable, but here's some example code on how to create an ordinal variable in R from scratch in general.

```
> ages <- c("18-24", "18-24", "45-54", "25-34",
"35-44")
> ages
[1] "18-24" "18-24" "45-54" "25-34" "35-44"
> factor_ages <- factor(ages, order = TRUE, levels =
c("18-24", "25-34", "35-44","45-54"))
> factor_ages
[1] 18-24 18-24 45-54 25-34 35-44
Levels: 18-24 < 25-34 < 35-44 < 45-54
```

Training the Proportional Odds Model

Now we will train our proportional odds model on the `soup` data set.

```
> install.packages(“MASS”)
> library(MASS)
> model_soup <- polr(SURENESS ~ PROD + DAY +
SOUPTYPE + SOUPFREQ + COLD + EASY + GENDER +
AGEGROUP + LOCATION, data=soup)
```

Done! Now we can take a look at the fitted model (next slide).

The Proportional Odds Model

```
> model_soup
Call:
polr(formula = SURENESS ~ PROD + DAY + SOUPTYPE + SOUPFREQ +
    COLD + EASY + GENDER + AGEGROUP + LOCATION, data = soup)

Coefficients:
         PRODTest               DAY2    SOUPTYPECanned    SOUPTYPEDry-mix
       1.17136760        -0.30165936       -0.28244222         0.21162826
SOUPFREQ1-4/month  SOUPFREQ<1/month            COLDYes              EASY2
      -0.06605588         0.02103962        0.27563555         0.08834772
            EASY3              EASY4             EASY5              EASY6
       0.24203251         0.22252688        0.16162041        -0.09887599
            EASY7              EASY8             EASY9             EASY10
      -0.03888040        -0.07270840       -0.04954610         0.90468338
     GENDERFemale      AGEGROUP31-40     AGEGROUP41-50      AGEGROUP51-65
      -0.02810994         0.03132997        0.22121004        -0.12787429
 LOCATIONRegion 2  LOCATIONRegion 3
      -0.09892899         0.04188272

Intercepts:
        1|2         2|3         3|4         4|5         5|6
-1.62143136 -0.62538979 -0.29570345 -0.03781062  0.63852579

Residual Deviance: 5329.491
AIC: 5383.491
```

The Proportional Odds Model

Finally, we can get our model's predicted probabilities for each observation lying in each class.

```
> predict(model_soup, data=soup, type="probs")
              1          2          3          4          5         6
1    0.19469920 0.20092809 0.08088198 0.06436448 0.15763297 0.3014933
2    0.06971162 0.09895791 0.05137623 0.04742228 0.15048781 0.5820442
3    0.19469920 0.20092809 0.08088198 0.06436448 0.15763297 0.3014933
4    0.06971162 0.09895791 0.05137623 0.04742228 0.15048781 0.5820442
5    0.24636375 0.22315929 0.08219809 0.06260173 0.14369746 0.2419797
```

In our paper (which I'll discuss next), Prof. Jacob Bien and I show that the proportional odds model does a better job of estimating the probabilities of each response lying in class 1 than a logistic regression model trained only to predict whether or not the response is in class 1.

So Far

So far, we've seen:

Estimating the probabilities of rare events precisely can be important.

Binary classifiers struggle in this setting (class imbalance).

Sometimes more common outcomes exist that are related to the rare outcome (on an ordinal scale.) Estimating the decision boundary associated with more common outcomes is easier.

If the common decision boundary gives us information about the rare decision boundary, then leveraging a more precise estimate of the common decision boundary leads to a better estimate the rare decision boundary.

Ultimately, this leads to better estimates of the probabilities of rare events.

Can we go beyond the proportional odds model?

Outline

1 Background and Practical Motivation

2 The Proportional Odds Model

3 Training the Proportional Odds Model

4 PRESTO

5 Data Application

Get these slides and code at https://github.com/gregfaletto/dataconla2023 (or scan the QR code)

A More Flexible Version of the Proportional Odds Model

Proportional odds model assumes that the $\boldsymbol{\beta}$ vector associated with each decision boundary is identical. We've seen that when an outcome is rare, this is an improvement over separate logistic regressions to estimate each decision boundary.

But it's still inflexible. **What if individual features have different influences on the decision boundaries at different levels?**

> In **online marketing**, it could be that some ad features make the ad "flashier" and increase the probability of a click, but are not predictive of the probability of purchasing.
>
> Students may place different weights on factors when **deciding whether to pursue graduate school vs. undergrad** (may have more appealing alternatives to additional schooling, may have different financial constraints, etc.)

Ideally, we'd like the more common decision boundaries to *inform* the estimation of the rare decision boundaries without imposing exact equality.

Proportional Odds Model Loss Function

Remember: the proportional odds model can be written as

$$\ln\left(\frac{\mathbb{P}\left(y \leq k \mid \boldsymbol{x}\right)}{\mathbb{P}\left(y > k \mid \boldsymbol{x}\right)}\right) = \alpha_k + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p, \qquad k = 1, \ldots, K-1.$$

The proportional odds model is typically estimated by this loss function:

$$\mathscr{L}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = -\sum_{i=1}^{n} \ln\left[\hat{\mathbb{P}}(y = y_i \mid \boldsymbol{x} = \boldsymbol{x}_i; \boldsymbol{\alpha}, \boldsymbol{\beta})\right],$$

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_{K-1})$ and $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_p)$ are the parameters to estimate, y_i is the label for observation i, $\boldsymbol{x}_i$ is the feature vector for observation i, and $\hat{\mathbb{P}}(y = y_i \mid \boldsymbol{x} = \boldsymbol{x}_i; \boldsymbol{\alpha}, \boldsymbol{\beta})$ is the model's estimated probability of observation i lying in its observed class y_i given its predictors $\boldsymbol{x}_i$ using the parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$.

Proportional Odds Model Loss Function

Proportional odds model is typically estimated by this loss function:

$$\mathscr{L}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = -\sum_{i=1}^{n} \ln\left[\hat{\mathbb{P}}(y = y_i \mid \boldsymbol{x} = \boldsymbol{x}_i; \boldsymbol{\alpha}, \boldsymbol{\beta})\right].$$

The negative of the $\ln(\,\cdot\,)$ function gets smaller as each $\hat{\mathbb{P}}(y = y_i \mid \boldsymbol{x} = \boldsymbol{x}_i; \boldsymbol{\alpha}, \boldsymbol{\beta})$ gets larger. So minimizing the loss function means finding parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ that make each $\hat{\mathbb{P}}(y = y_i \mid \boldsymbol{x} = \boldsymbol{x}_i; \boldsymbol{\alpha}, \boldsymbol{\beta})$ as large as possible (maximum likelihood).

This is sometimes called a **cross-entropy** loss function. The logistic regression loss function can be written in this way when there are $K = 2$ responses. This is also a standard loss function for multiclass classification problems.

PRESTO!

Proportional odds model is typically estimated by this loss function:

$$\mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = -\sum_{i=1}^{n} \ln\left[\hat{\mathbb{P}}(y = y_i \mid \boldsymbol{x} = \boldsymbol{x}_i; \boldsymbol{\alpha}, \boldsymbol{\beta})\right].$$

Unlike the proportional odds model, PRESTO learns a different $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_{K-1}$ for each decision boundary. But PRESTO imposes an ℓ_1 **penalty** on the differences between coefficients corresponding to the same feature in adjacent decision boundaries:

$$\mathcal{L}(\boldsymbol{\alpha}, \boxed{\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_{K-1}}) = -\sum_{i=1}^{n} \ln[\hat{\mathbb{P}}(y = y_i \mid \boldsymbol{x} = \boldsymbol{x}_i; \boldsymbol{\alpha}, \boxed{\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_{K-1}})] + \boxed{\lambda \sum_{j=1}^{p} \left(\left|\beta_{j1}\right| + \sum_{k=2}^{K-1} \left|\beta_{jk} - \beta_{j,k-1}\right| \right)}.$$

PRESTO!

$$\mathscr{L}(\boldsymbol{\alpha}, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_{K-1}) = -\sum_{i=1}^{n} \ln\left[\hat{\mathbb{P}}(y = y_i \mid \boldsymbol{x} = \boldsymbol{x}_i; \boldsymbol{\alpha}, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_{K-1})\right] + \lambda \sum_{j=1}^{p} \left(\left|\beta_{j1}\right| + \sum_{k=2}^{K-1} \left|\beta_{jk} - \beta_{j,k-1}\right| \right)$$

The ℓ_1 penalty encourages coefficients for adjacent decision boundaries to be similar—the loss function is smaller if all of the $\left|\beta_{jk} - \beta_{j,k-1}\right|$ terms equal exactly 0. If this happens, we return to the proportional odds model.

In this way, the loss function allows decision boundaries with abundant data to influence the estimated rare decision boundaries, just like the proportional odds model.

But PRESTO has flexibility to allow differences in decision boundaries if the observed data suggests it would help enough to “overrule” the penalty (improvement over proportional odds).

PRESTO!

$$\mathscr{L}(\boldsymbol{\alpha}, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_{K-1}) = -\sum_{i=1}^{n} \ln\left[\hat{\mathbb{P}}(y = y_i \mid \boldsymbol{x} = \boldsymbol{x}_i; \boldsymbol{\alpha}, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_{K-1})\right] + \lambda \sum_{j=1}^{p} \left(\left|\beta_{j1}\right| + \sum_{k=2}^{K-1} \left|\beta_{jk} - \beta_{j,k-1}\right| \right)$$

The λ penalty tuning parameter is chosen by cross-validation, just like in the lasso.

The direct penalization of the $\left|\beta_{j1}\right|$ terms (the parameters that define the first decision boundary) acts as regularization that improves overall estimation of the model.

PRESTO is inspired by the **fused lasso** (Tibshirani et. al. 2005): places an ℓ_1 (lasso) penalty not on the coefficients themselves, but on differences between adjacent coefficients.

PRESTO

In brief, PRESTO estimates something like the proportional odds model, but allows the coefficient vectors for each decision boundary to be different.

However, it regularizes the differences.

The differences in the coefficient corresponding to the same feature at adjacent decision boundaries are penalized: $|\beta_{j,k} - \beta_{j,k-1}|$

If all of these differences equal 0, we have the proportional odds model. If none of them equal 0, we have completely free decision boundaries.

Penalization encourages these differences to be 0 unless the data "overrules" this because allowing the coefficients to differ is "worth it" for predictive performance.

PRESTO learns from the data which parts of the common decision boundaries can be used to inform estimation of the rare decision boundaries, without the rigid proportional odds assumption.

PRESTO

For more details on PRESTO, read our paper at https://proceedings.mlr.press/v202/faletto23a.html.

You can also get the code for our paper (including code implementing PRESTO) at https://github.com/gregfaletto/presto.

Summary

Binary classifiers struggle to estimate rare probabilities due to class imbalance.

If there are ordinal outcomes, a decision boundary with abundant data nearby can be leveraged to improve estimation of rare decision boundary (and rare probabilities),

Proportional odds model allows this, but imposes exactly equality of $\boldsymbol{\beta}$ vectors (inflexible).

PRESTO relaxes proportional odds, allowing $\boldsymbol{\beta}$ vectors to differ but imposing ℓ_1 penalty on differences.

This allows for best of both worlds: learn from abundant decision boundaries, but flexibly adapt for different decision boundaries between different outcomes.

Outline

1 Background and Practical Motivation

2 The Proportional Odds Model

3 Training the Proportional Odds Model

4 PRESTO

5 Data Application

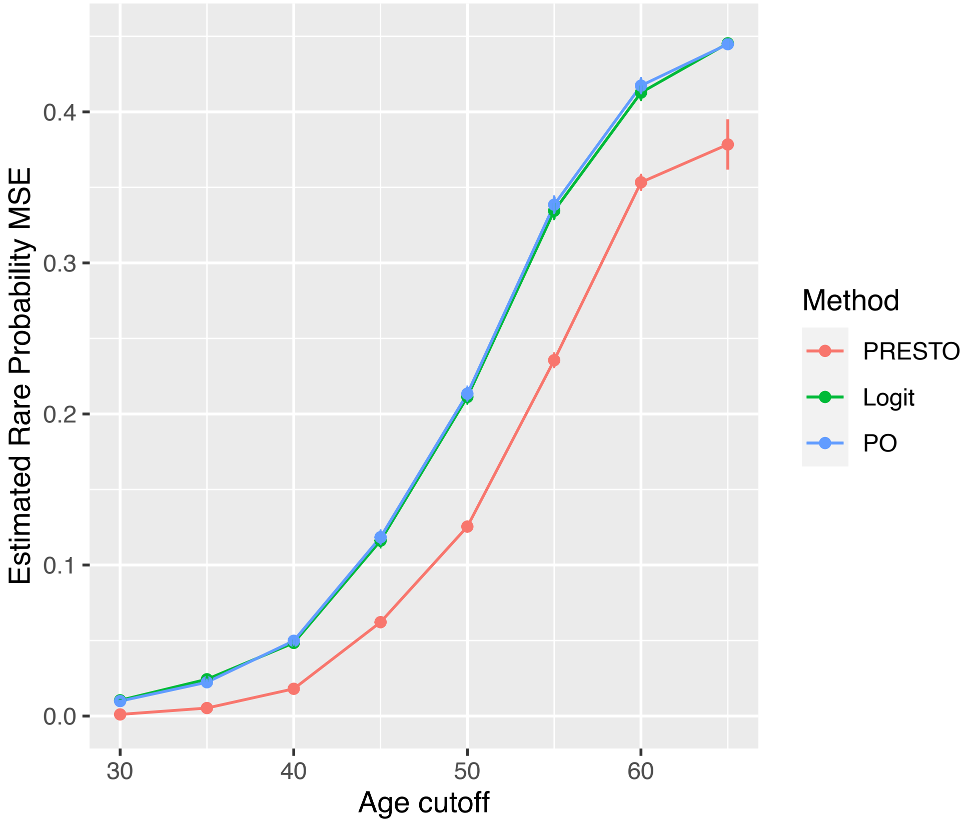
Get these slides and code at https://github.com/gregfaletto/dataconla2023 (or scan the QR code)

Data Application

Data set: 3059 patients who were eventually diagnosed with diabetes. For each patient: age they were diagnosed with prediabetes, age diagnosed with diabetes, predictors.

For each age cutoff, create an ordered response variable: by this age, did the patient have prediabetes, diabetes, or neither?

Use 90% of the data to fit a model, hold out 10% for testing. Use each model to predict whether or not patient had diabetes at this age on test set. Do this 35 times for each age.

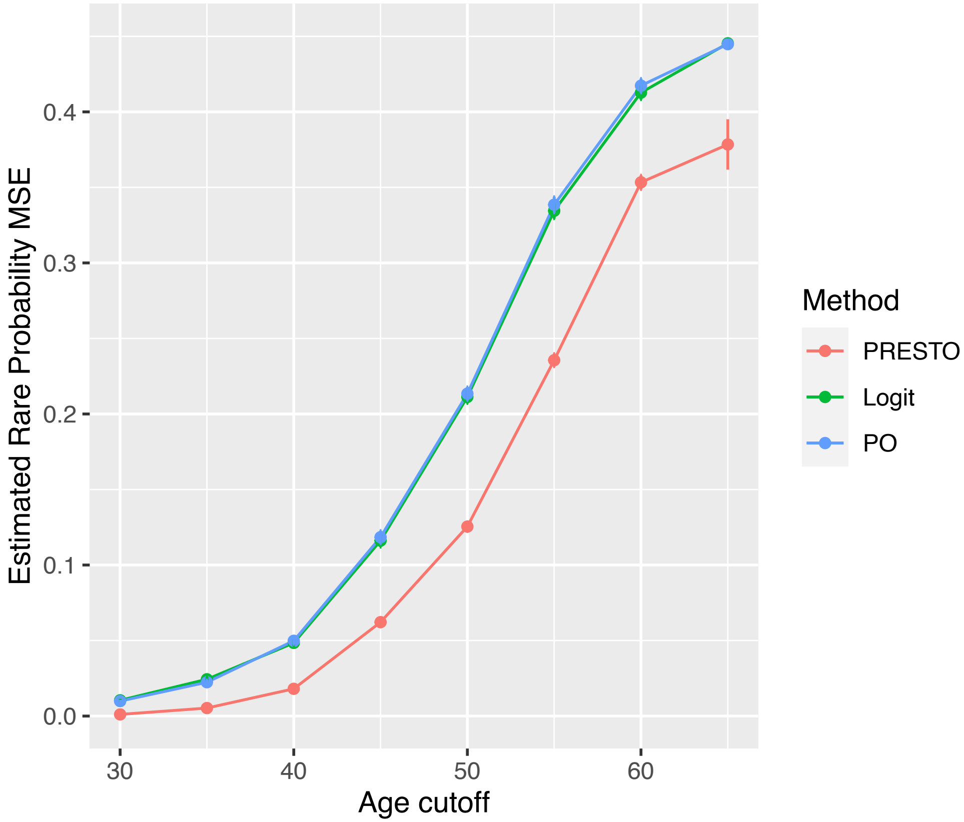


Data Application

Methods used: PRESTO, logistic regression (solely predicting whether the patient has diabetes), proportional odds model

PRESTO outperforms both other methods.

> Logistic regression fails to learn from the more common cases of patients with prediabetes.
>
> Proportional odds is too rigid, can’t adapt to a different decision boundary for patients with diabetes.



Get these slides,
code, and more

Thank you!

References

G. Faletto & J. Bien (2023). Predicting Rare Events by Shrinking Towards Proportional Odds. *Proceedings of the 40th International Conference on Machine Learning*, in *Proceedings of Machine Learning Research* 202:9547-9602 Available from https://proceedings.mlr.press/v202/faletto23a.html.

G. James, D. Witten, T. Hastie, & R. Tibshirani (2021). *An introduction to statistical learning with applications in R* (Vol. 112, p. 18). New York: Springer.

P. McCullagh (1980). Regression models for ordinal data. J*ournal of the Royal Statistical Society: Series B (Methodological)*, 42(2):109–127.

R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight (2005). Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108.