Feature Engineering

Data Science & Machine Learning Team

02/22/2021

Andrew Wheeler, PhD

andrew.wheeler@hms.com

Agenda

- Machine Learning and Causality
  - Importance of business domain knowledge

- Functional form
  - Polynomial terms, non-linear effects, step functions
  - Encoding categorical variables

- Examples in Python using linear regression

Machine Learning and Causality

- To get a better prediction, need a basic understanding of the causal mechanisms behind the phenomenon
  - Need to feed the machine the correct data in the correct format, or it will not generate valid predictions in practice.
- Example business problems at HMS
  - Subrogration – younger people are more likely to have car accidents
  - Payment Integrity – some insurance claim types are more discretionary, more likely to be upcoded
  - Technical Denial – some claims have higher complexity, and so are likely to be missing critical information
- Each requires unique solutions – same model architecture would not work for all three projects

What is Feature Engineering?

- A model has inputs used to predict an output, e.g.:

$$y = f(a, b, c)$$

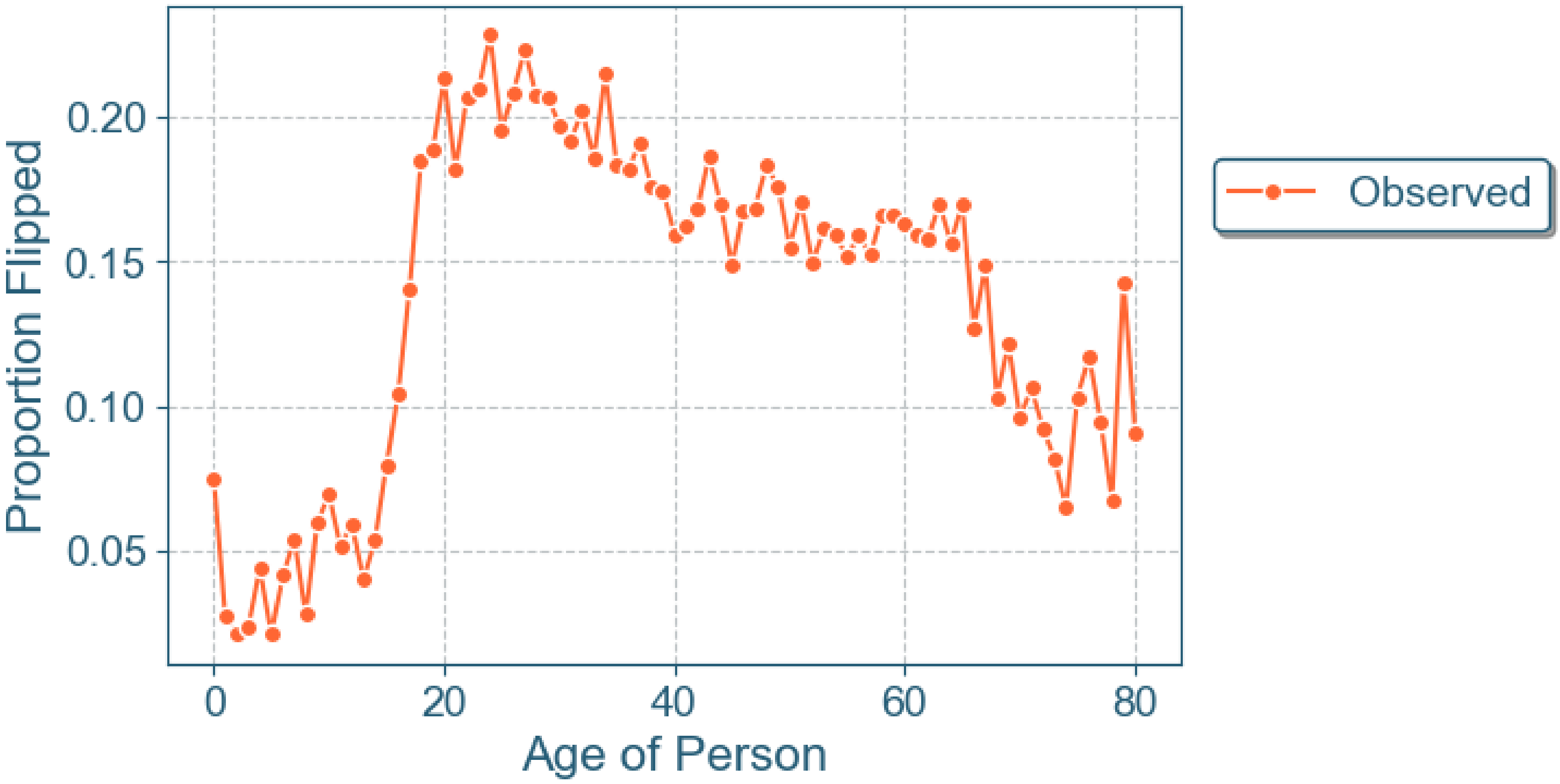
- Feature engineering involves:
  - What features to include in the model, e.g. (a,b,c) instead of (a,e,f)?
  - Transformations of variables, e.g. $\log(y) = \beta_1 \cdot \sqrt{a}$
  - Representing categorical variables in a model, e.g.

$$y = \beta_1(b = \text{Aetna}) + \beta_2(b = \text{Amerihealth})$$

  - Combinations of all of these variables together, e.g. interaction effects

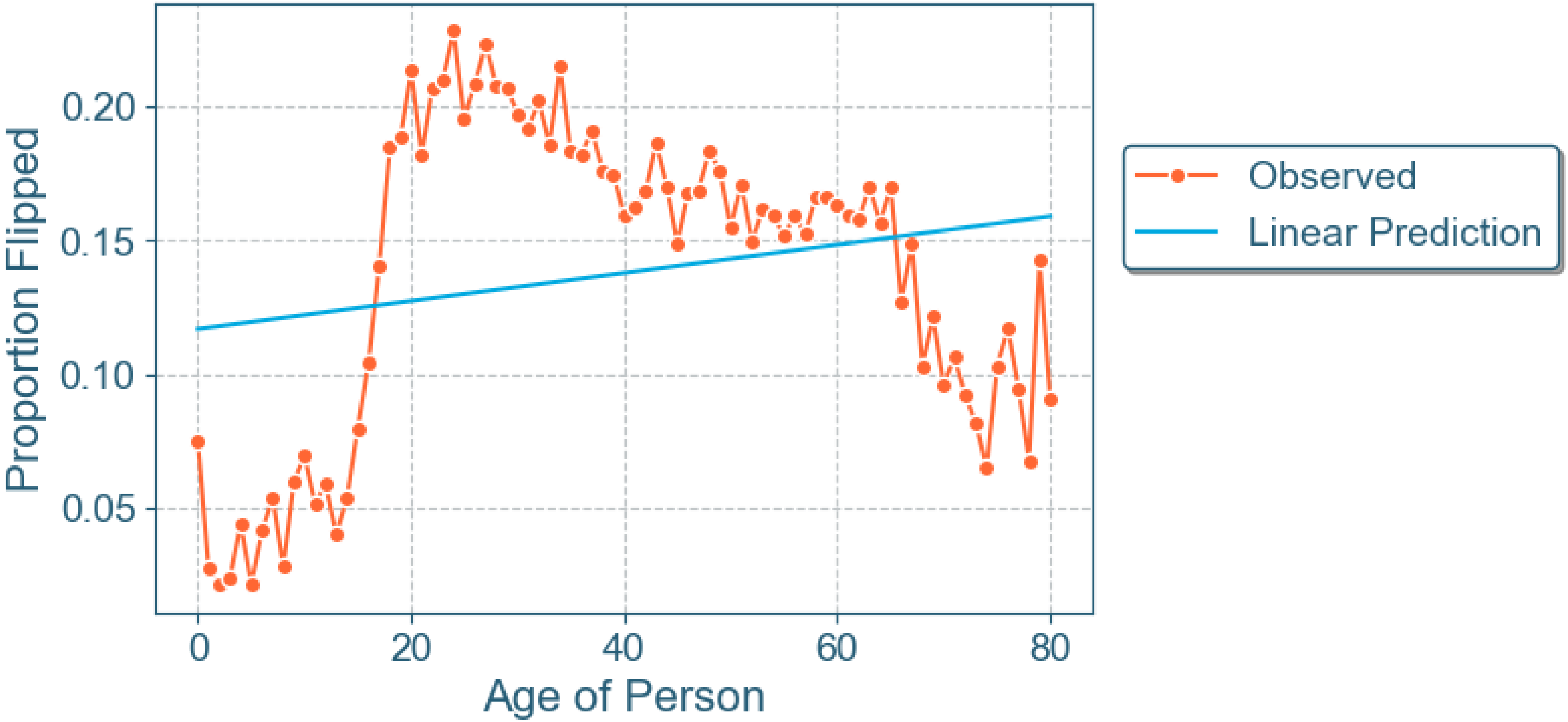
$$y = \beta_1\sqrt{a} + \beta_2\,(b = \text{Aetna}) + \beta_3(b = \text{Amerihealth}) + \beta_4(\sqrt{a} \cdot [b = \text{Aetna}])$$

Example: Subrogation (via Accent)

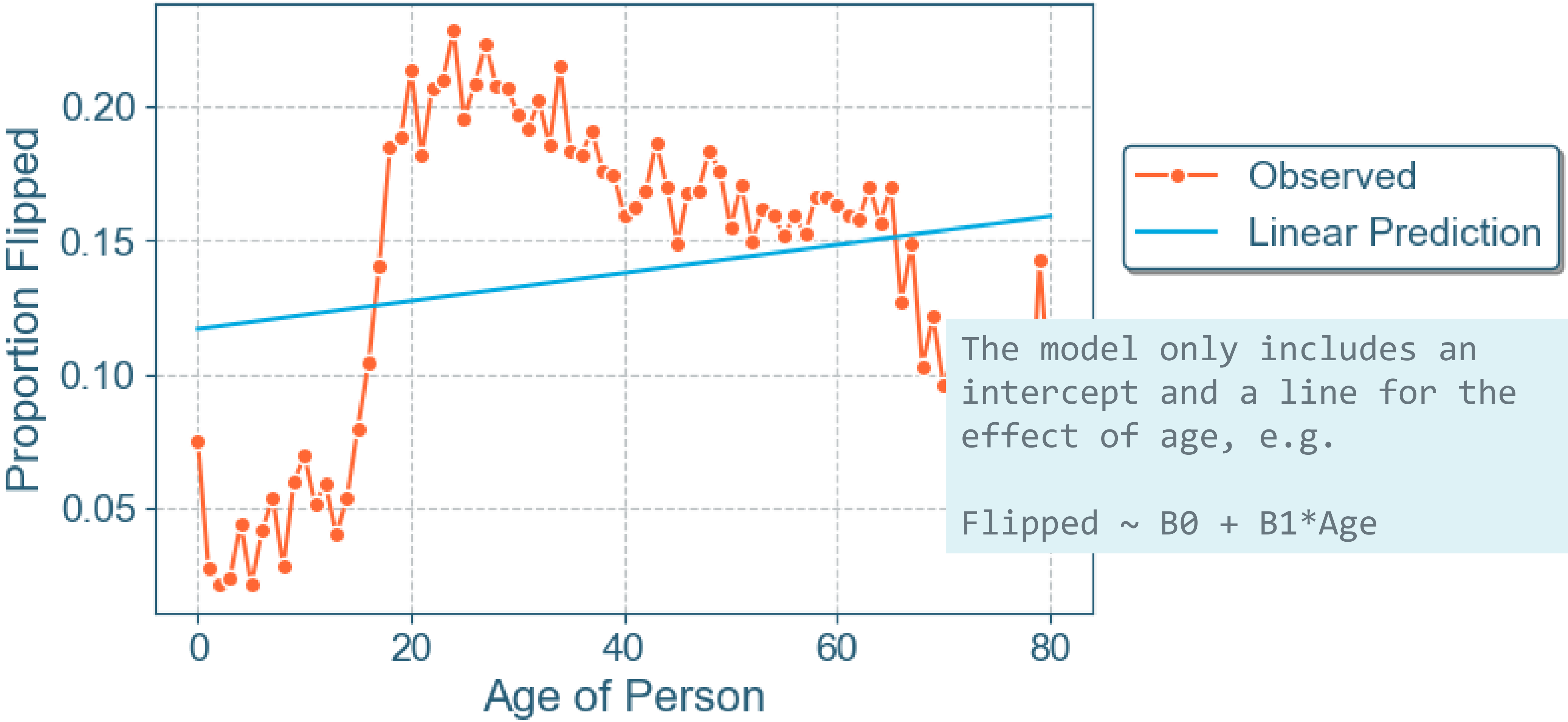


	Age	PercentFlipped
Index		
0	0	0.075314
1	1	0.027778
2	2	0.021786
3	3	0.023641
4	4	0.044118
...	...	...
76	76	0.117188
77	77	0.094862
78	78	0.067568
79	79	0.142857
80	80	0.090909

Example: Subrogation (via Accent) – Linear



Example: Subrogation (via Accent) – Linear

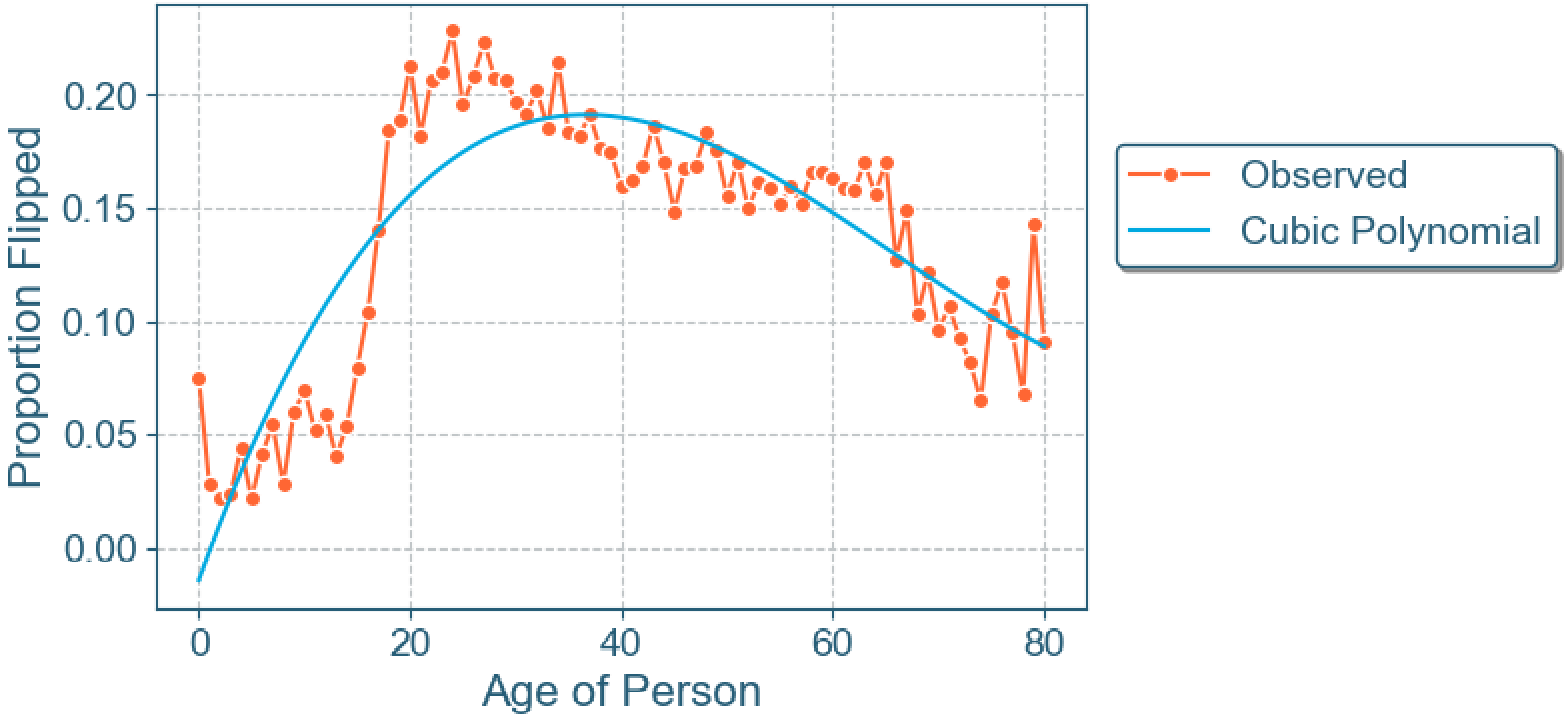


The model only includes an intercept and a line for the effect of age, e.g.

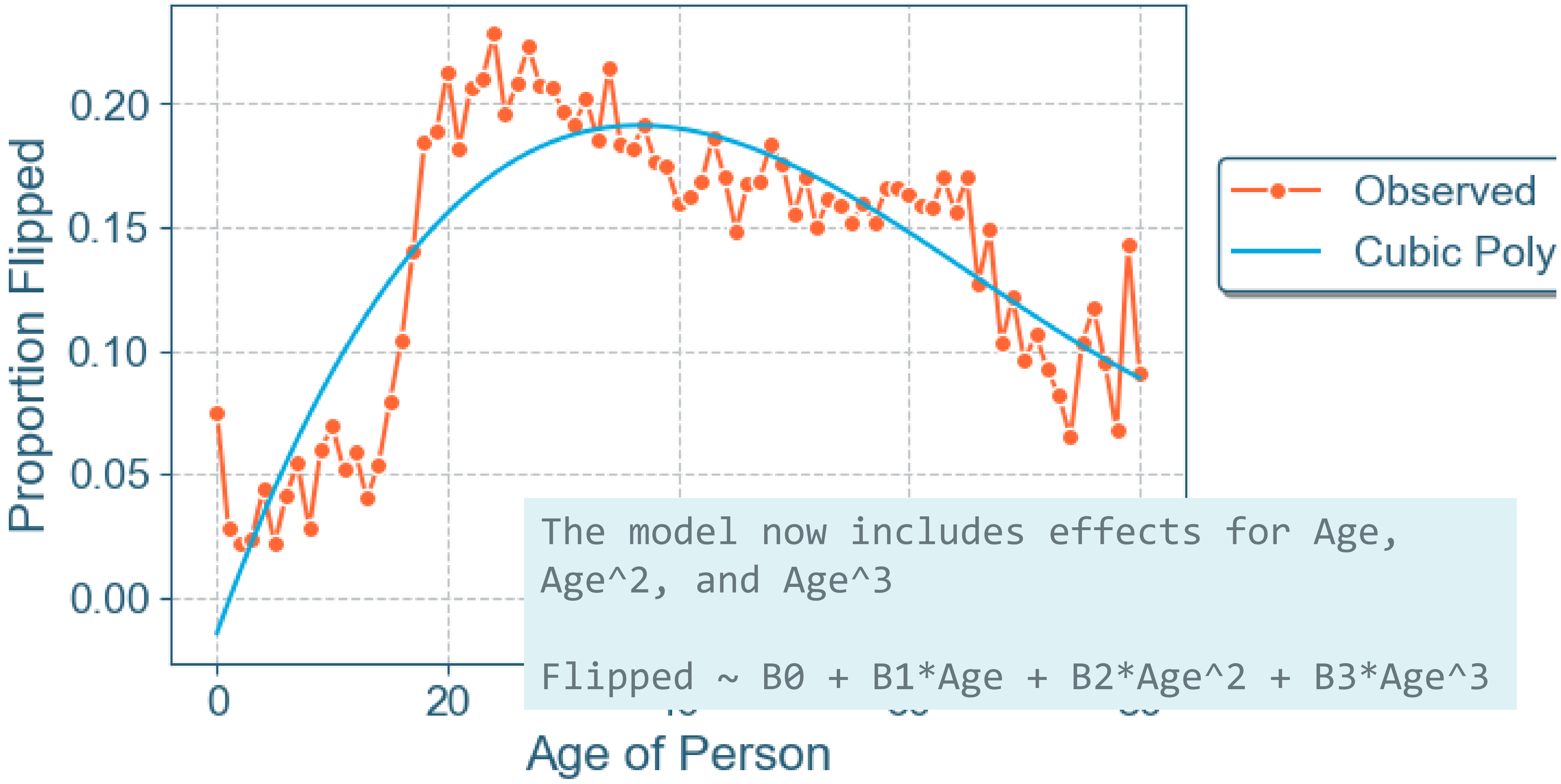
Flipped ~ B0 + B1*Age

	Age
Index	
0	0
1	1
2	2
3	3
4	4
...	...
76	76
77	77
78	78
79	79
80	80

Example: Subrogation (via Accent) – Polynomial



Example: Subrogation (via Accent) – Polynomial

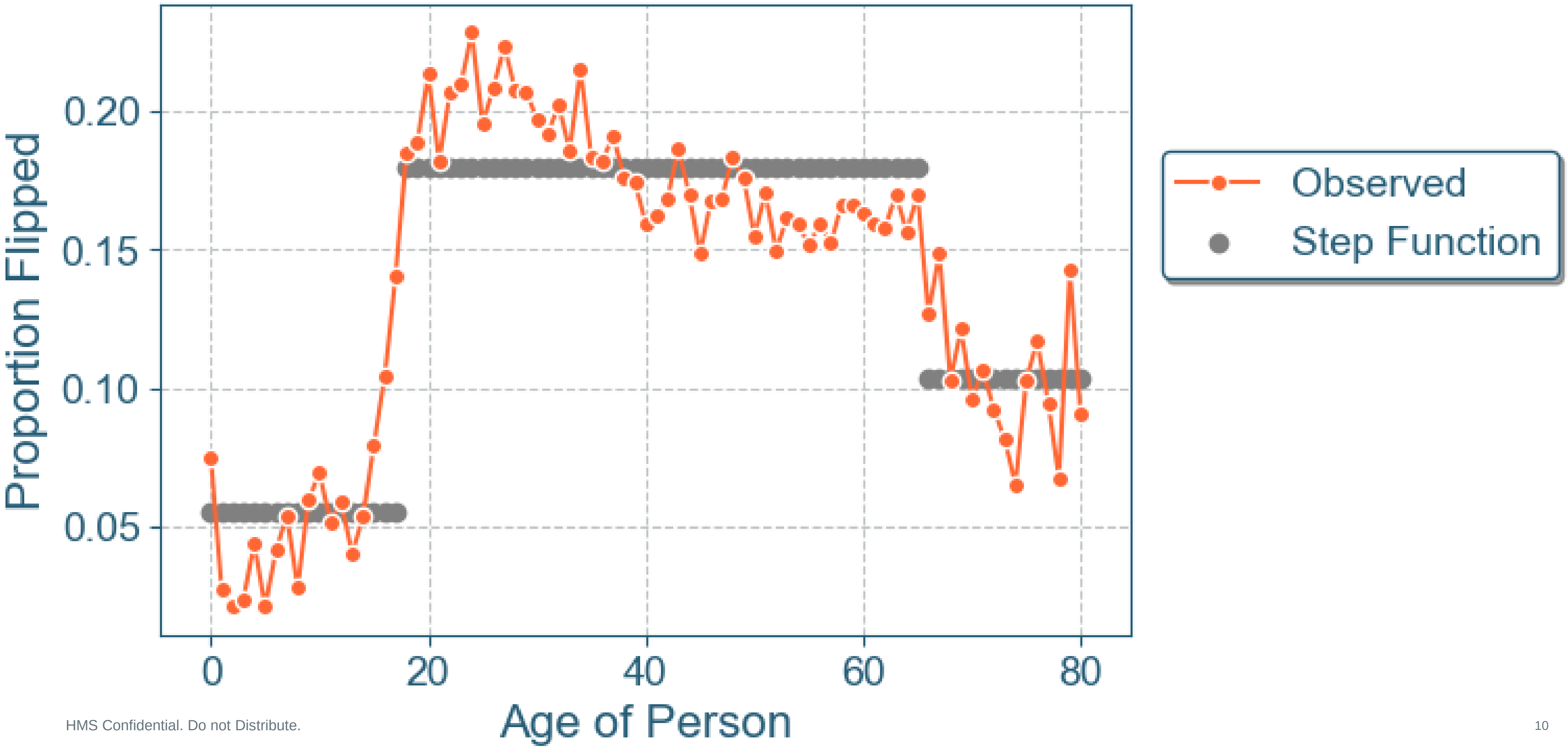



```
The model now includes effects for Age,
Age^2, and Age^3

Flipped ~ B0 + B1*Age + B2*Age^2 + B3*Age^3
```

Index	Age	AgeSq	AgeCu
0	0	0	0
1	1	1	1
2	2	4	8
3	3	9	27
4	4	16	64
...	...	...	...
76	76	5776	438976
77	77	5929	456533
78	78	6084	474552
79	79	6241	493039
80	80	6400	512000

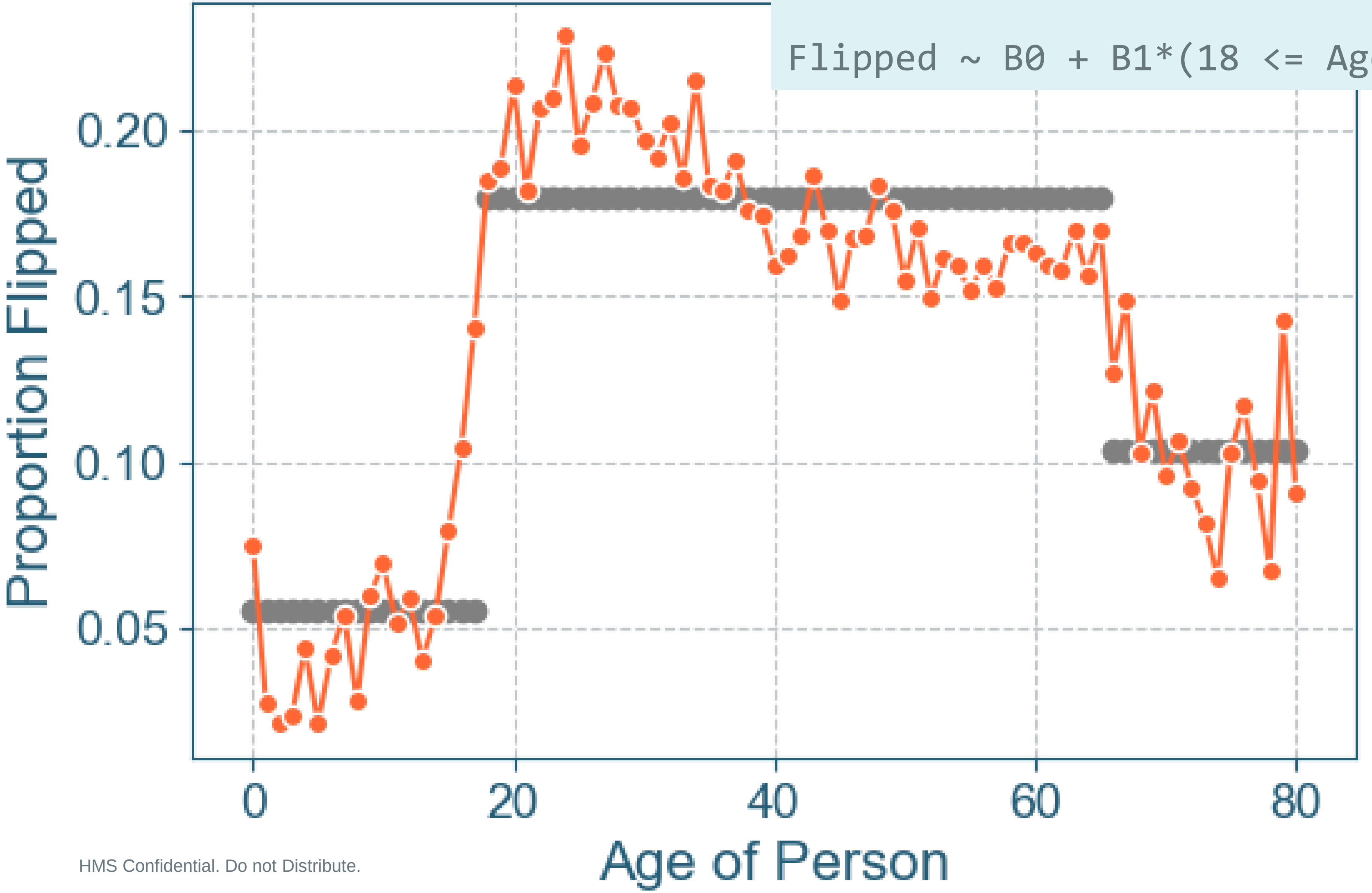
Example: Subrogation (via Accent) – Steps



Example: Subrogati

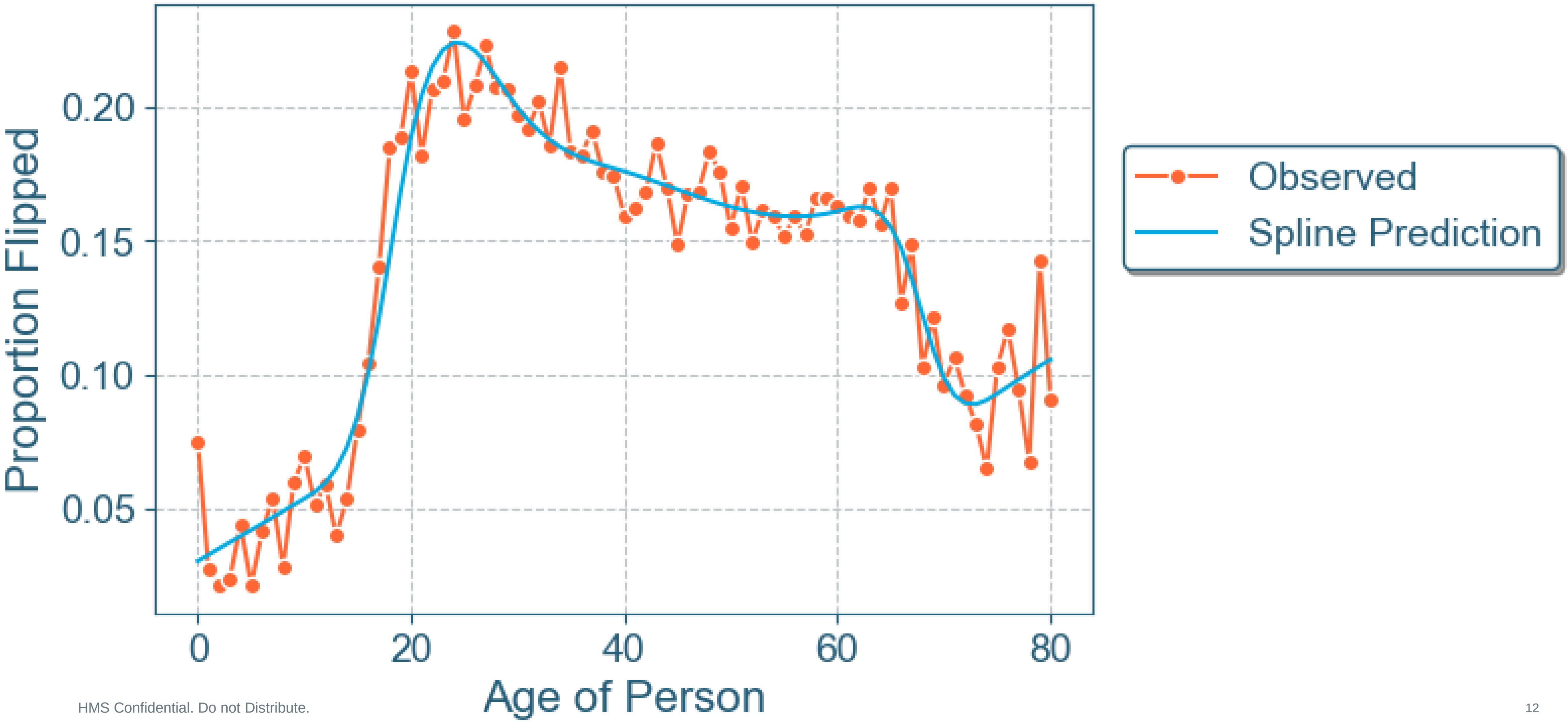
```
The model now is a series of step functions based on
age.

Flipped ~ B0 + B1*(18 <= Age =< 65)  + B2*(Age > 65)
```

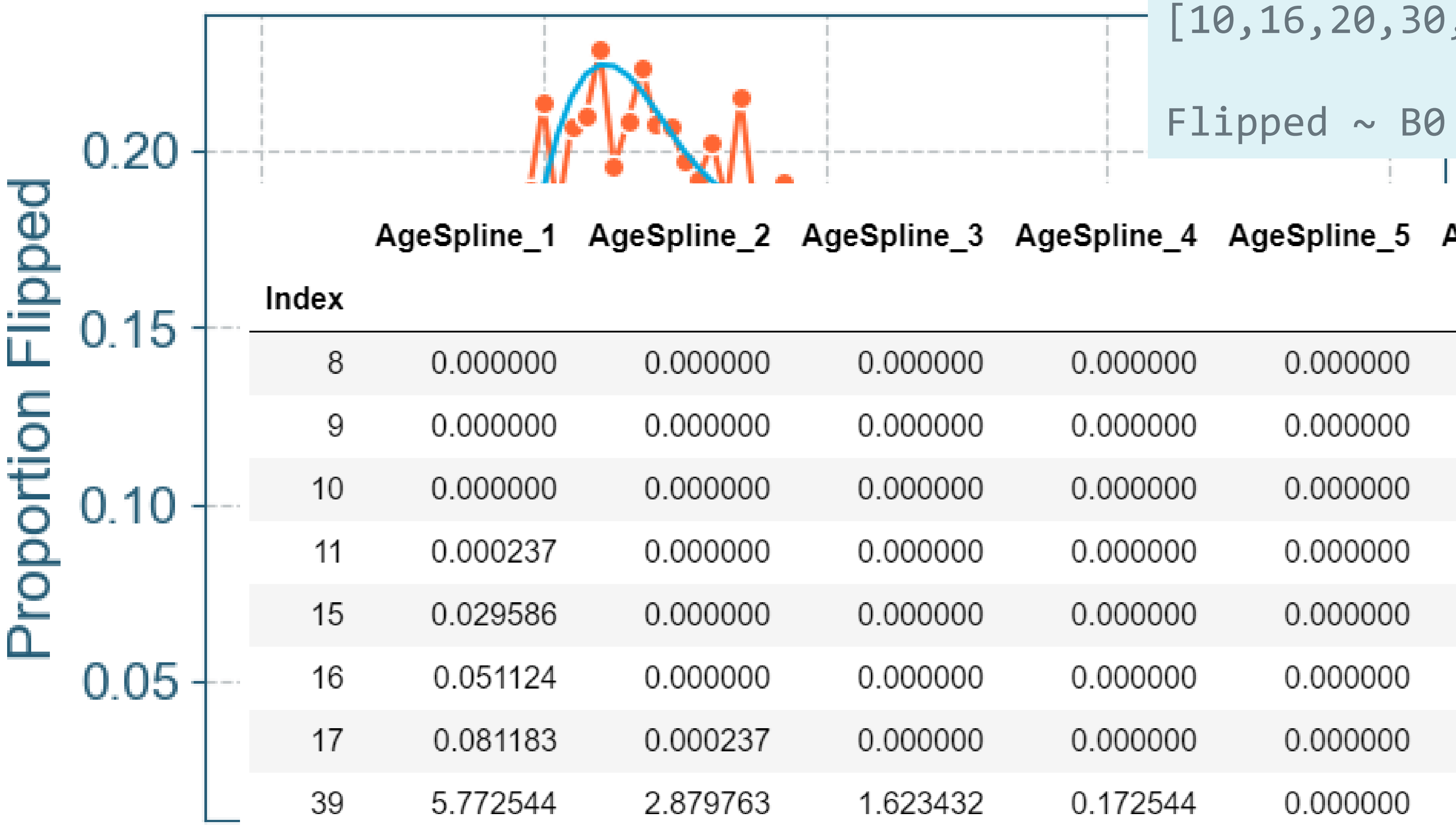


Index	Age18_65	AgeOver65
16	0	0
17	0	0
18	1	0
19	1	0
20	1	0
63	1	0
64	1	0
65	1	0
66	0	1
67	0	1

Example: Subrogation (via Accent) – Splines



Example: Subrogation (via A




```
Splines are more complicated functions,
but similar to polynomials. This example
includes knots at
[10,16,20,30,40,50,60,66,70,75]

Flipped ~ B0 + B1*Age + sum[ Bk*s(Age) ]
```

Index	AgeSpline_1	AgeSpline_2	AgeSpline_3	AgeSpline_4	AgeSpline_5	AgeSpline_6	AgeSpline_7	AgeSpline_8	Age
8	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.0	8
9	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.0	9
10	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.0	10
11	0.000237	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.0	11
15	0.029586	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.0	15
16	0.051124	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.0	16
17	0.081183	0.000237	0.000000	0.000000	0.000000	0.0	0.0	0.0	17
39	5.772544	2.879763	1.623432	0.172544	0.000000	0.0	0.0	0.0	39
40	6.390533	3.271953	1.893491	0.236686	0.000000	0.0	0.0	0.0	40
41	7.051124	3.698225	2.191953	0.315030	0.000237	0.0	0.0	0.0	41

Encoding Categorical Variables

- For low numbers of categories, one-hot encoding, or dummy variables, is the best you can do:

Sex
Male
Female
Unknown

Male	Female	Unknown
1	0	0
0	1	0
0	0	1

- For many categories, will have another lecture on *High Cardinality*.
  - What counts as many? Depends on data size, typically want a 100+ observations for the smallest category though.
  - For many categories, may be impossible to fit in memory

Other Example Feature Engineering Ideas

- Interaction effects between different variables,
  - Age*Male + Age*Female
  - Age*Client1 + Age*Client2, etc.

- Ratio effects, e.g. for loans debt/income ratio, for claims netpaid/length of stay

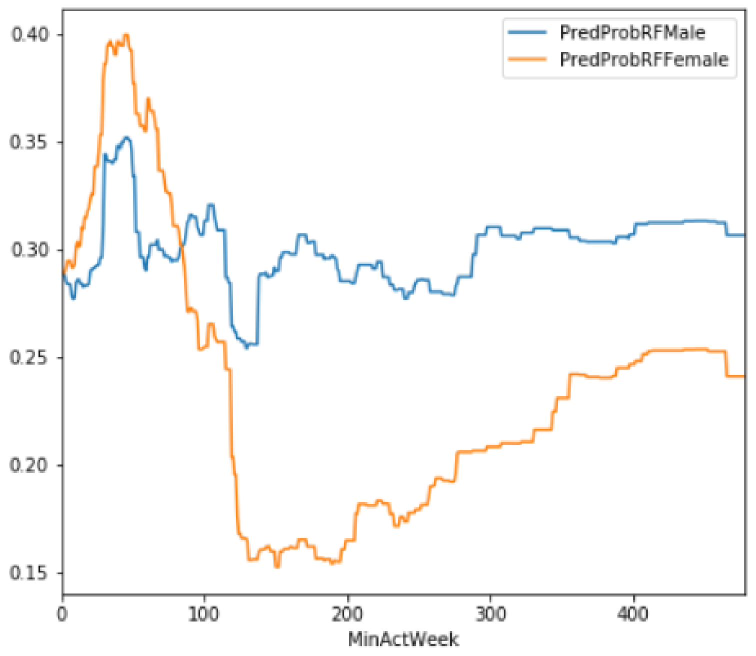
- Relative to a group, e.g. [actual netpaid - average netpaid per DRG] or [observed length of stay – average length of stay per DRG]

Different ML models and FE

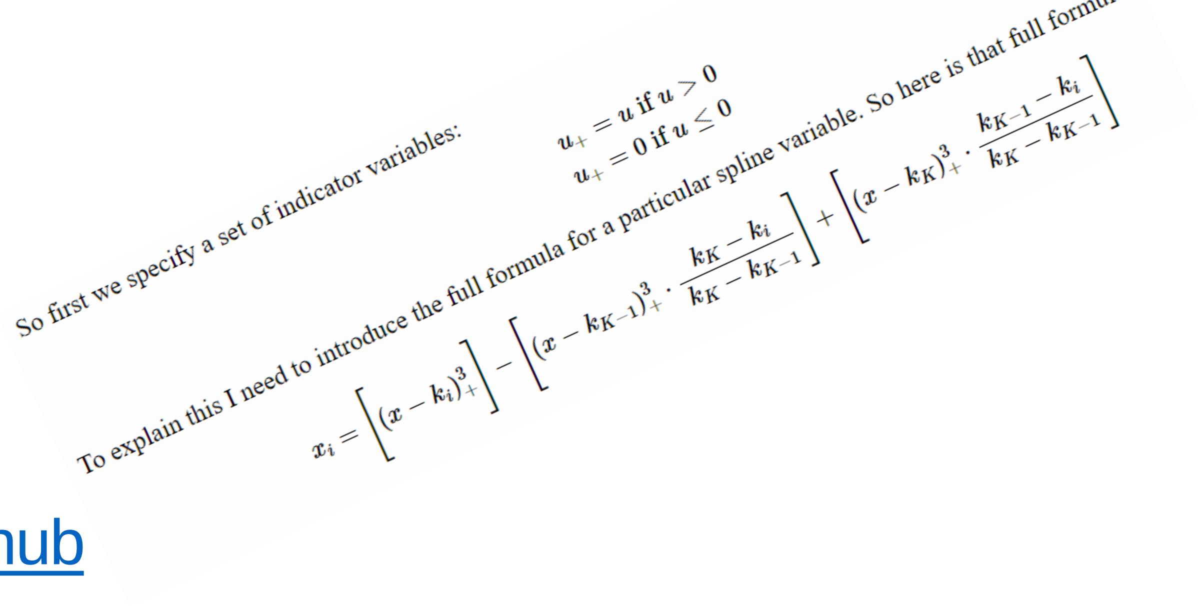
- Forest based models are very good at finding non-linear effects, so don't necessarily need to include non-linear terms (like spline or polynomial variables)

- But …. they can often improve model fit, especially if they are important variables

- It is ok to include related features, e.g. log(netpaid) and netpaid, in the same *predictive* model



Other Resources



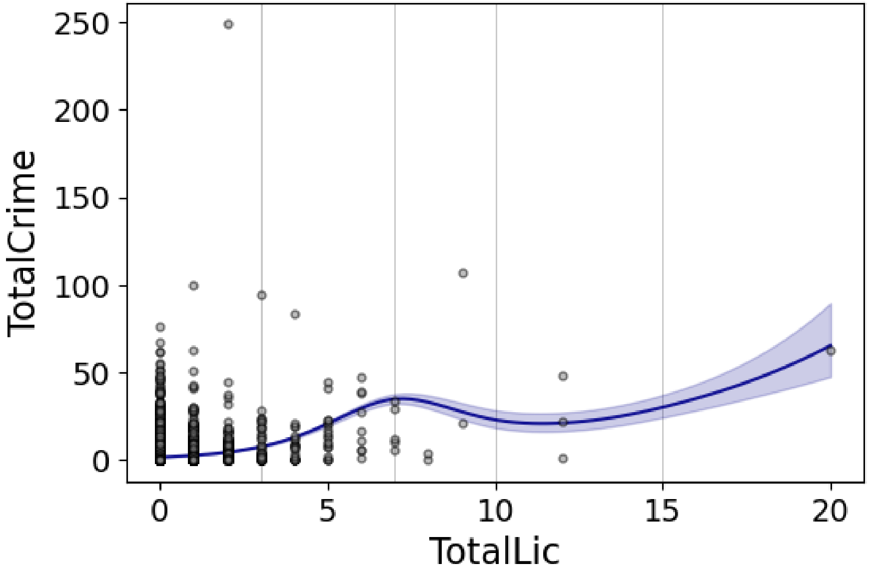
So first we specify a set of indicator variables:

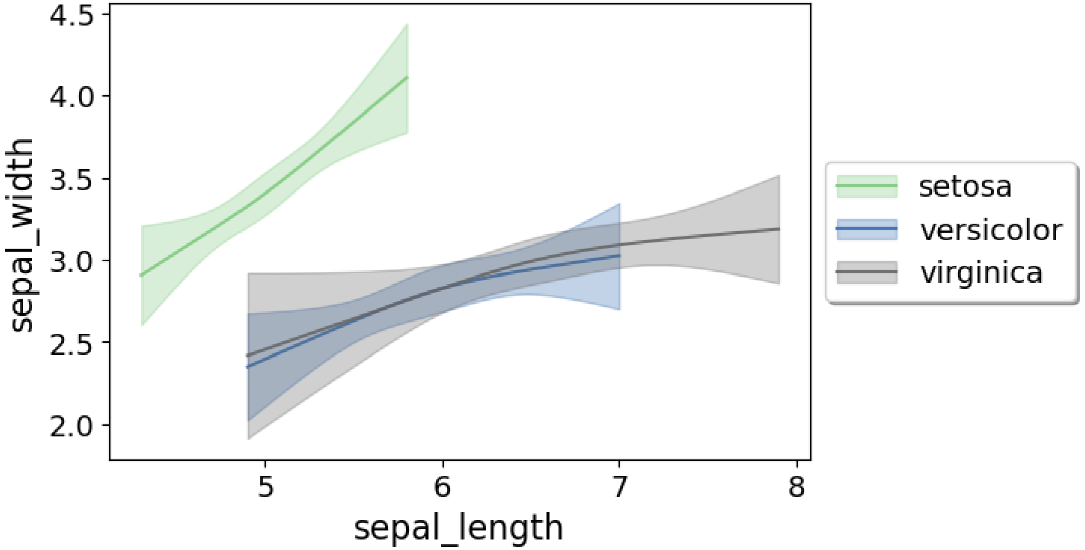
$$u_+ = u \text{ if } u > 0$$
$$u_+ = 0 \text{ if } u \leq 0$$

To explain this I need to introduce the full formula for a particular spline variable. So here is that full formu

$$x_i = \left[(x - k_i)_+^3\right] - \left[(x - k_{K-1})_+^3 \cdot \frac{k_K - k_i}{k_K - k_{K-1}}\right] + \left[(x - k_K)_+^3 \cdot \frac{k_{K-1} - k_i}{k_K - k_{K-1}}\right]$$

- Notes on Restricted Cubic Splines
- Notebook with these examples on Github
- Smooth.py functions to plot exploratory relationships





Future Topics

- Dealing with a high number of categories in models
- Feature Importance metrics for predictive models
- Partial dependence plots to understand functional form
- Reduced form interpretable machine learning summaries

Questions?

Future Topics

Have requests?

Let me know!

Introduction to Data Science Course Outline

Andrew Wheeler, PhD, andrew.wheeler@hms.com

- Lesson 01: Data Science 101
- Lesson 02: Machine Learning 101
- Lesson 03: Evaluating Predictions
- Lesson 04: Intro Data Transformation in Python
- Lesson 05: Data Visualization 101
- Lesson 06: Feature Engineering
- Lesson 07: Missing Data
- Lesson 08: Big Data and Parallel Computing Intro
- Lesson 09: Dimension Reduction and Unsupervised Learning
- Lesson 10: High Cardinality (Many Categories)
- Lesson 11: Intro to Forecasting
- Lesson 12: Conducting Experiments

Feature Engineering

Data Science & Machine Learning Team

02/22/2021

Andrew Wheeler, PhD

andrew.wheeler@hms.com