Modeling concepts, hyperparameter tuning, and telematics

June 2020

Today’s agenda

- Review some modeling concepts
- Intro to XGBoost
- Hyperparameter optimization
- Telematics loss modeling best practices

Parts of a model

Parts of a Model

- Scoring formula
- Objective function
- Optimization process
- Data

Parts of a Model

Scoring formula

- Ordinary least squares regression $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2$
- Generalized linear model $y = g^{-1}(\beta_0 + \beta_1 x_1 + \beta_2 x_2)$
- GLM with log link function $y = e^{\beta_0} \times e^{\beta_1 x_1} \times e^{\beta_2 x_2}$
- Decision tree model

$$
\begin{aligned}
&\text{If } x_1 < \beta_1 \text{ then } y = \mu_0 \\
&\text{If } x_1 \geq \beta_1 \text{ then: If } x_2 < \beta_2 \text{ then } y = \mu_1 \\
&\qquad\qquad\qquad\quad \text{If } x_2 \geq \beta_2 \text{ then } y = \mu_2
\end{aligned}
$$

Parts of a Model

Objective function

- What is the goal of the model?
- “Best model” needs to be well defined
- Must define the measure and the direction

- Ordinary Least Squares regression -> minimize mean squared error
- Generalized Linear Model -> maximize likelihood
- Decision tree -> minimize Gini impurity

Parts of a Model

Optimizing process

This is where the math can get complex!

Differential calculus, numerical methods, matrix mathematics may be employed.

GLM (and ordinary least squared regression)

-> Maximum likelihood estimation

-> Iteratively Reweighted Least Squares – much faster, doesn't estimate likelihoods

Neural nets

-> Gradient descent

Data Partitions

Data Partitions

Train, Validate and Test Data

Data scientists frequently split their data into 3 subsets: Train, Validate, Test

: For determining parameters estimates (ie, “train” the model). Train

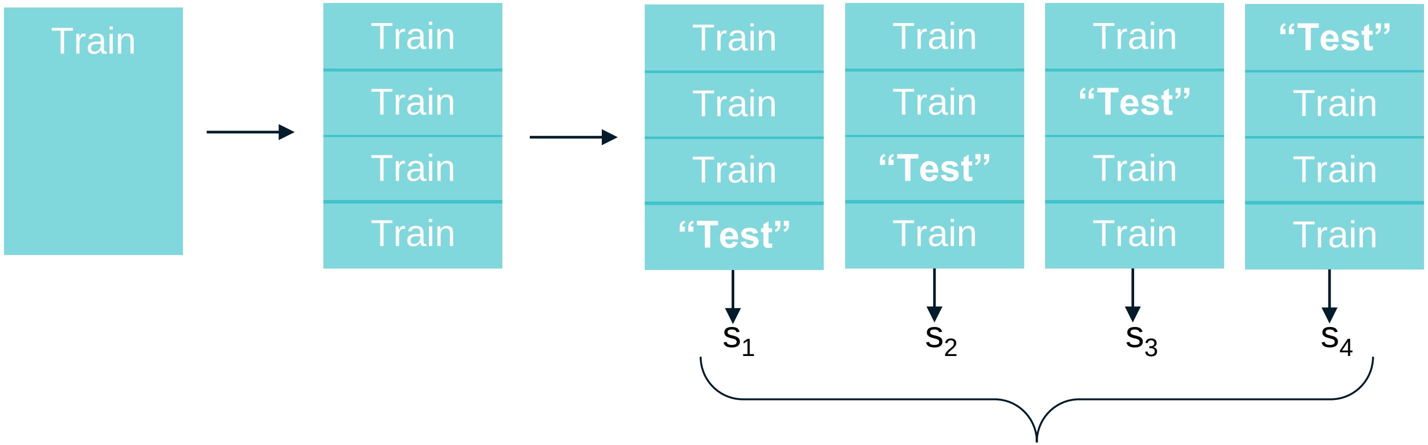
: For tuning hyperparameters to improve model performance Validate

: “Untouched” data to provide an unbiased evaluation of final model Test

Data Partitions

Cross-Validation

- Another approach is to split data into Train and Test
- Use cross-validation for tuning hyperparameters



$$(S_1 + S_2 + S_3 + S_4)/4 = \hat{S}$$

Hyperparameters

What is a hyperparameter?

- Scoring formula for GLM with log link function

$$y = e^{\beta_0} \times e^{\beta_1 x_1} \times e^{\beta_2 x_2}$$

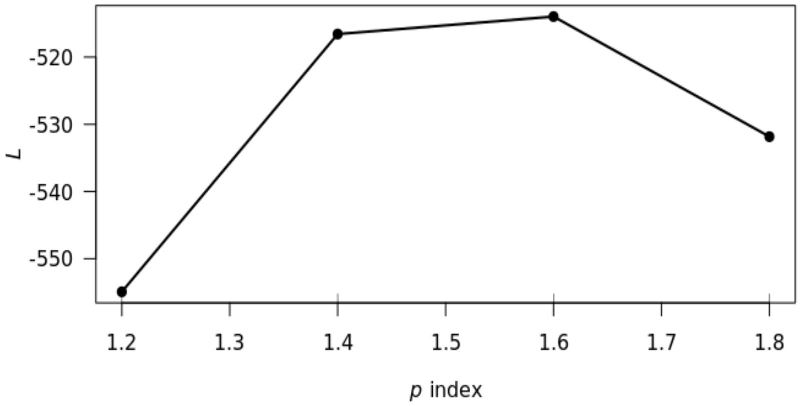
- β_0, β_1 and β_2 are parameters

- Hyperparameters are parameters outside of the scoring formula that affect
  - Model performance
  - Model complexity

Tweedie Power Parameter

- Tweedie GLMs have a hyperparameter called the Tweedie power parameter (p)
- Tweedie power parameter is important because it affects deviance, which affects the significance of variables in the model
- We can find the optimal value of p by testing different values of p to find the greatest likelihood

```r
set.seed=11
test.data <-rtweedie(n=200, power=1.46,mu=260,phi=100)
tweedie.profile(test.data ~ 1,
                p.vec=seq(1.2, 1.8, by=0.2),
                do.plot=TRUE)
```

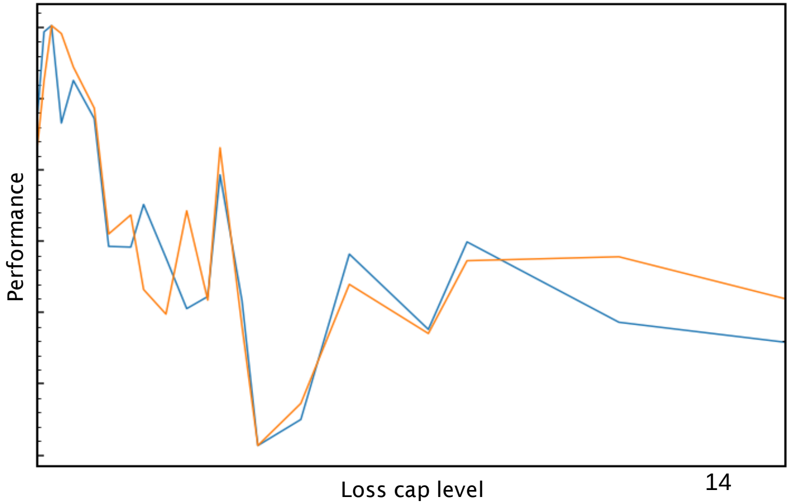


Hyperparameters

Loss capping

- Higher layers of loss can be uninformative due to their volatility
- Capping losses at the optimal level can improve model accuracy on uncapped losses
- The large loss threshold is a hyperparameter

- I used the grid search method to find a good level to cap losses
- I used 5-fold cross-validation to measure 2 different objective functions to measure how good each loss capping levels performed
- After determining where to cap losses, I estimated the Tweedie power parameter.



XGBoost

What is XGBoost?

- “eXtreme Gradient Boosting”

- “The algorithm of choice for many winning teams of machine learning competitions” according to the XGBoost website

- Why actuaries and data scientists may be interested in XGBoost:
  - Supports Tweedie, Poisson and gamma objective functions
  - Credibility-like parameter shrinkage
  - Finds predictive complex interactions
  - Automated variable selection

XGBoost Attributes

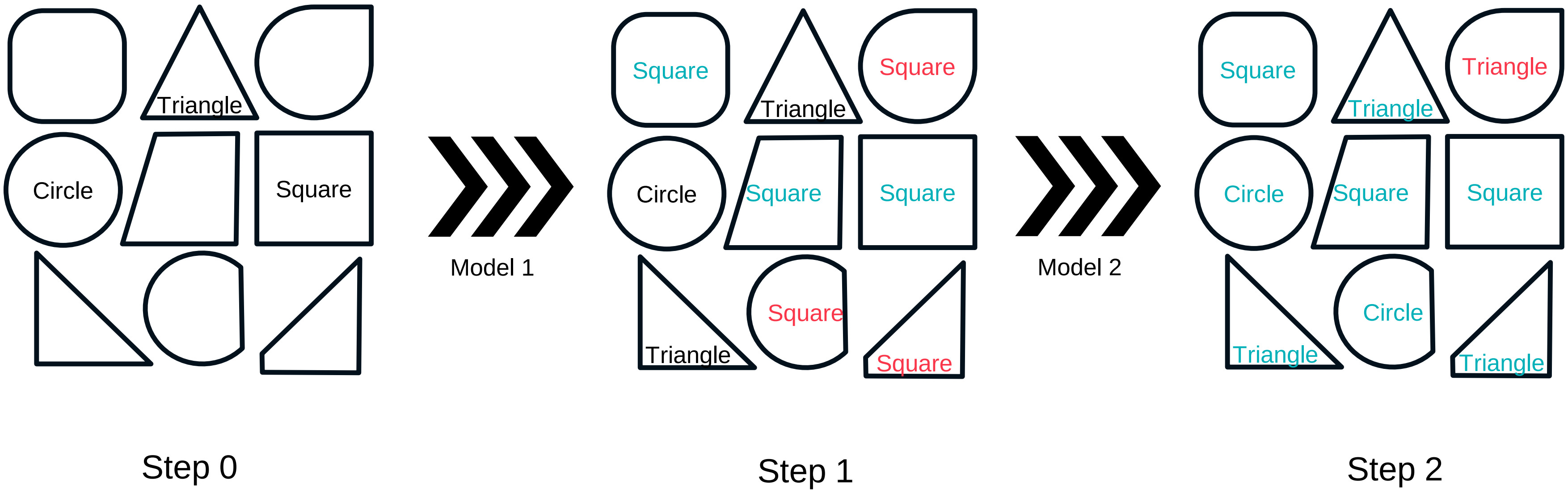
- Scoring formula comprised of simple If-Then-Else statements
  - ```
    Prediction = 0
    If v1 <B1             then add e1 to prediction
    If v1>=B1 and v2< B2 then add e2 to prediction
    ```

- Objective functions include

○ squared error	○ logistic	○ poisson		○ L1 regularization
○ squared log error	○ aft	○ gamma	+	○ L2 regularization
○ hinge	○ pairwise	○ tweedie		

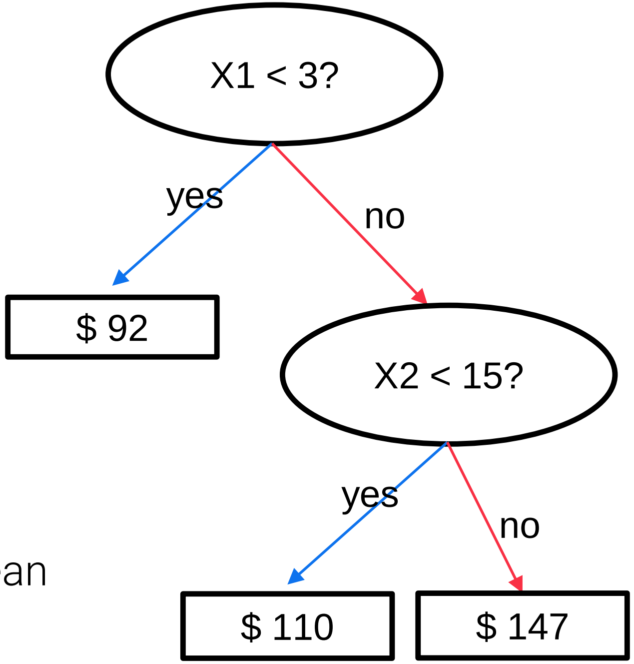
- Optimization function
  - For creating the trees, Recursive Binary Splitting is default, and other options available
  - Various options for boosting
    ```

What is Boosting?



A few important hyperparameters in XGBoost

- eta – Learning rate. Prevents overfitting
- gamma – Minimum loss reduction to make a split
- max_depth – Maximum depth of trees
- alpha – L1 regularization removes weaker variables
- lambda – L2 regularization shrinks estimates closer to the mean

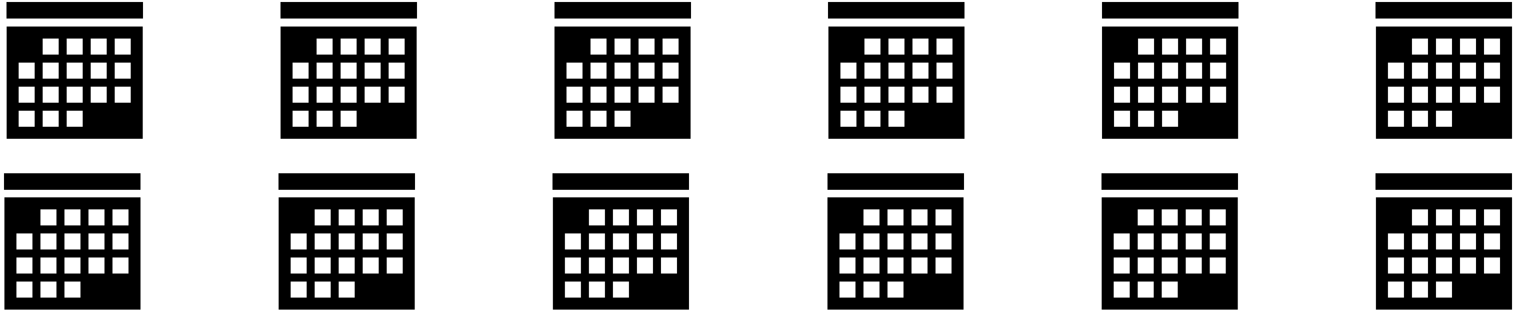


Tuning Hyperparameters

Tuning Hyperparameters

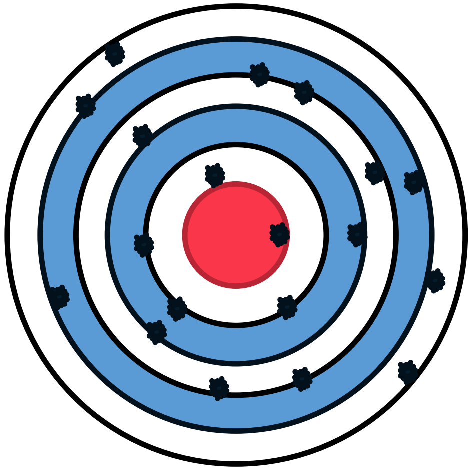
Grid Search

- We could use Grid search
  - If we use 10 values for each of the 5 parameters
  - Grid search would run the regression 100,000 times
  - If the model takes 5 minutes to converge then grid search will take 347 days of compute time
  - Grid search allocates a lot of effort to explore every combination of hyperparameters



Random Search

- We could use Random search
  - In many instances, random search performs about as well as grid search



- Random Search approach
  1. Select reasonable ranges for selected hyperparameters
  2. Randomly select combinations of values for hyperparameters
  3. Test combination
  4. Repeat

Tuning Hyperparameters

Example Tuning

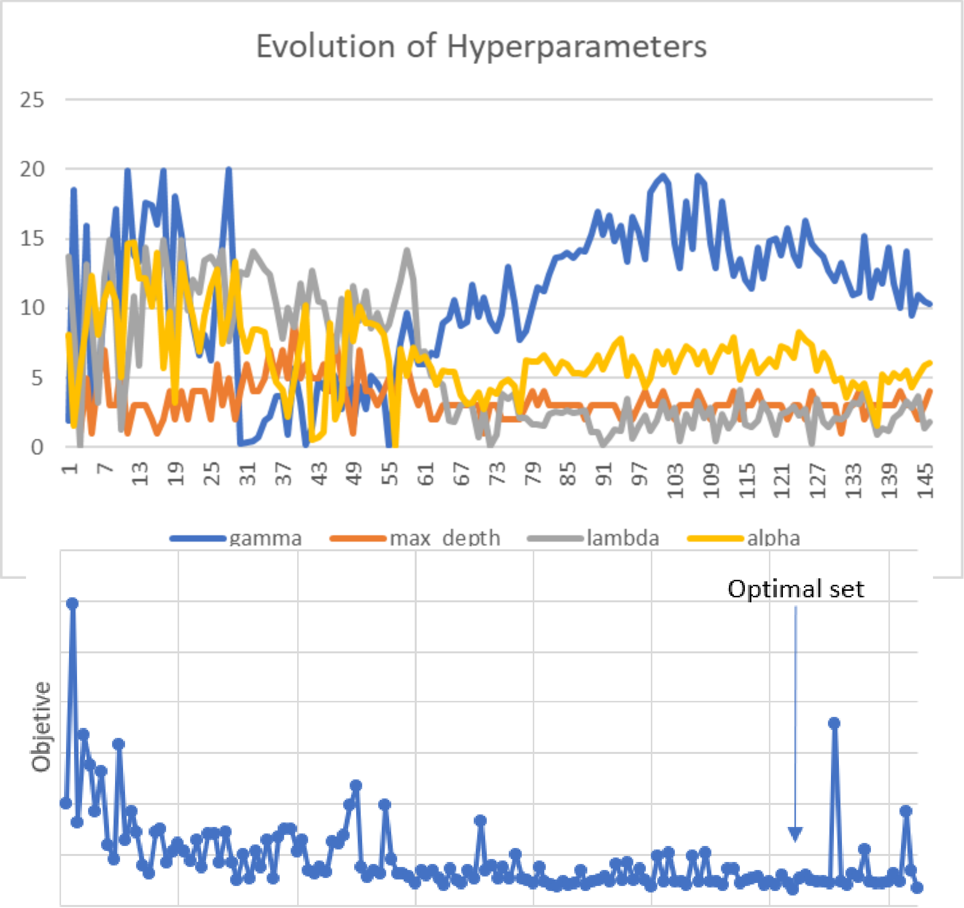
- For a recent project, I tuned 8 hyperparameters for an XGBoost model

Top graph

- Four of the hyperparameters are plotted
- x-axis = trial number
- y-axis = hyperparameter values

Bottom

- x-axis = trial number
- y-axis = objective function value



Takeaways

Summary

- Algorithms like XGBoost have features that actuaries will be interested in using:
  - Supports Tweedie, Poisson and gamma objective functions
  - Credibility-like parameter shrinkage
  - Finds predictive complex interactions
  - Automated variable selection

- Hyperparameter tuning is important for some algorithms
  - Easy to understand methods for hyperparameter tuning exist
  - Some more advanced algorithms are very complex and can find optimal hyperparameters much faster than simpler methods

Telematics loss modeling

Telematics Loss Modeling Best Practices

Predicts future insurance loss

Accounts for traditional factors

Developed using credible data

Customer friendly

PREDICTS FUTURE INSURANCE LOSS

Defining risk variables iteratively

Generate hypothesis

- Logic
- Telematics expertise

IMPLEMENTATION

Collect behaviors

- Prototype, pilot, and deploy data capture

ANALYSES

Evaluate loss correlation

- Merge with Insurance data
- Define variables

DEPLOYMENT

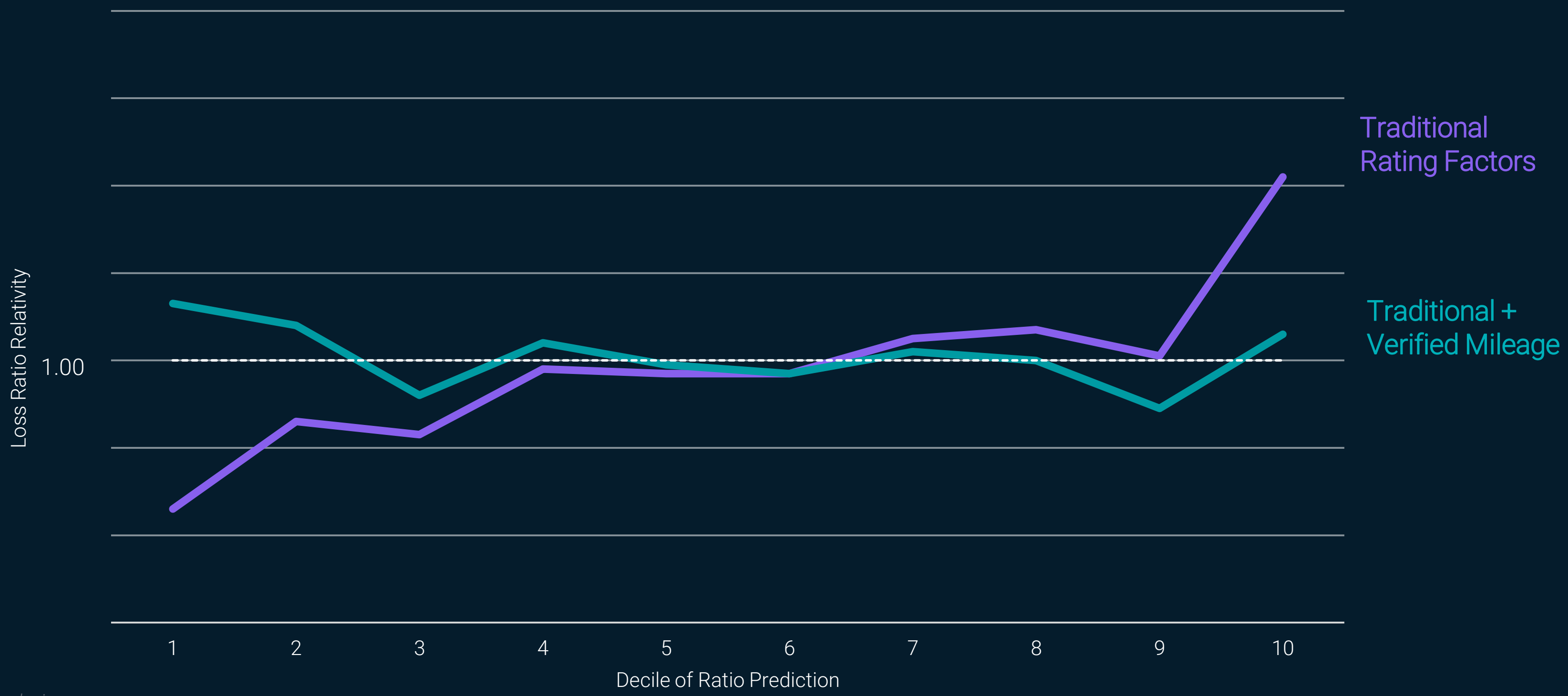
Develop score and file

- Predictive models
- Support regulators

FEEDBACK FROM STAKEHOLDERS

PREDICTS FUTURE INSURANCE LOSS

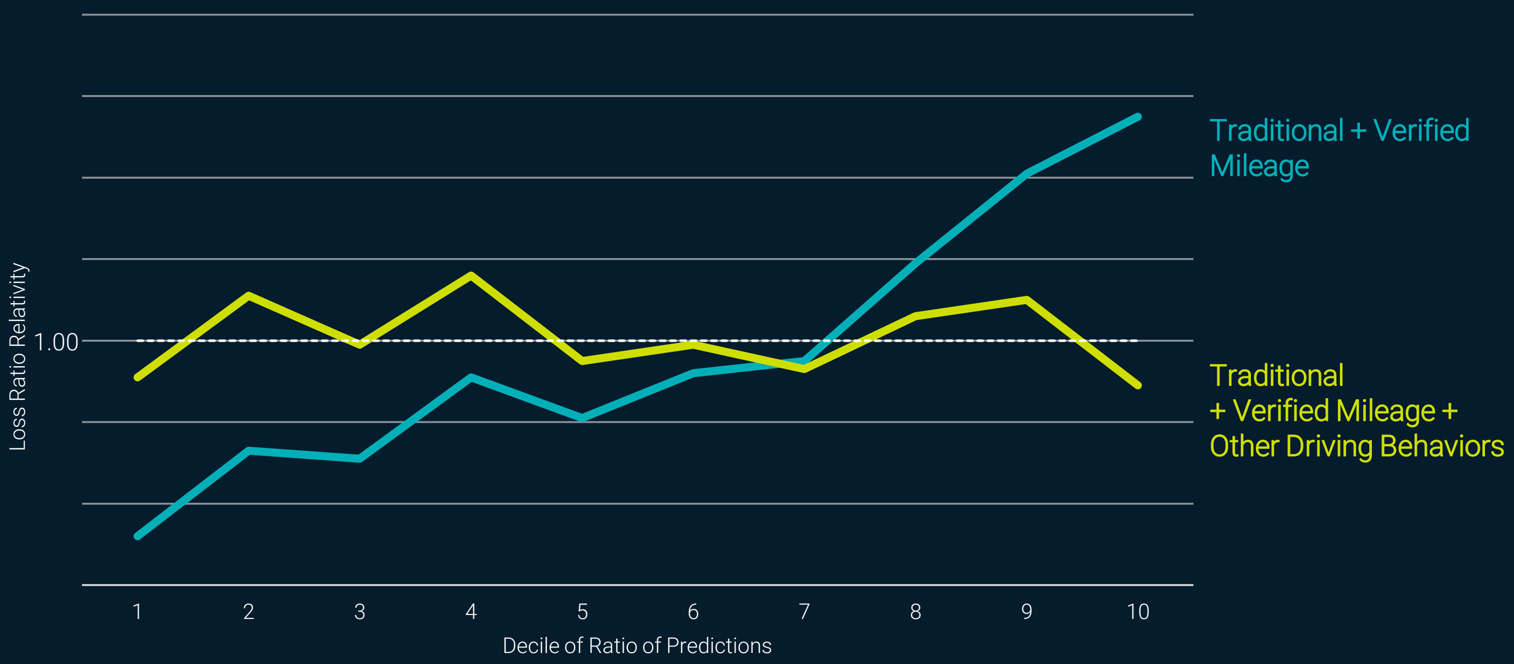
Mileage is good



Source: Arity analysis

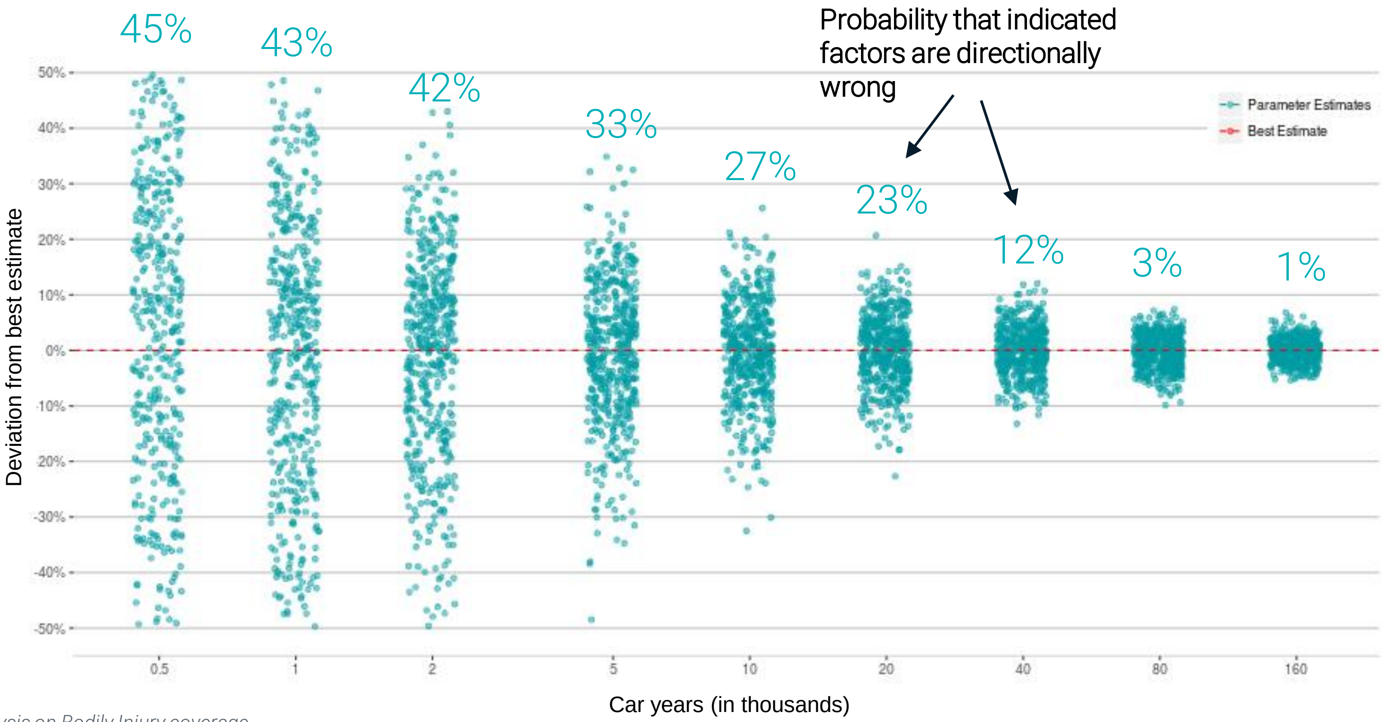
PREDICTS FUTURE INSURANCE LOSS

Behavior is better



Source: Arity analysis

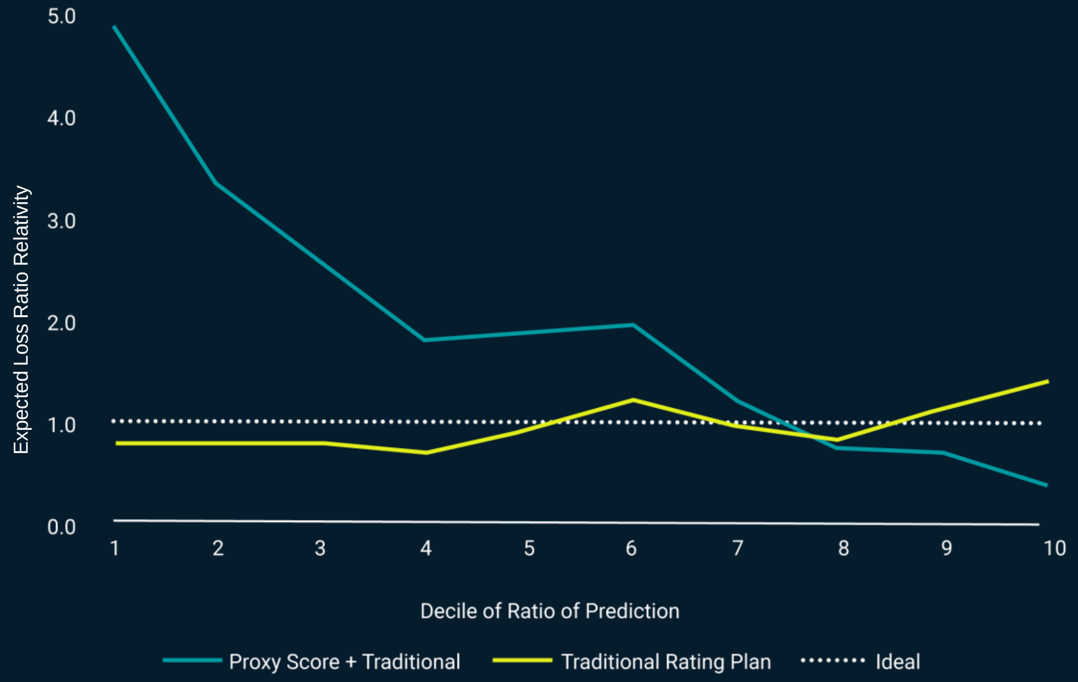
Data volume, accuracy, and duration of collection matter



Source: Arity analysis on Bodily Injury coverage

PREDICTS FUTURE INSURANCE LOSS

Insights built on proxied collisions could make the rating plan worse



Source: Arity analysis

Thank You