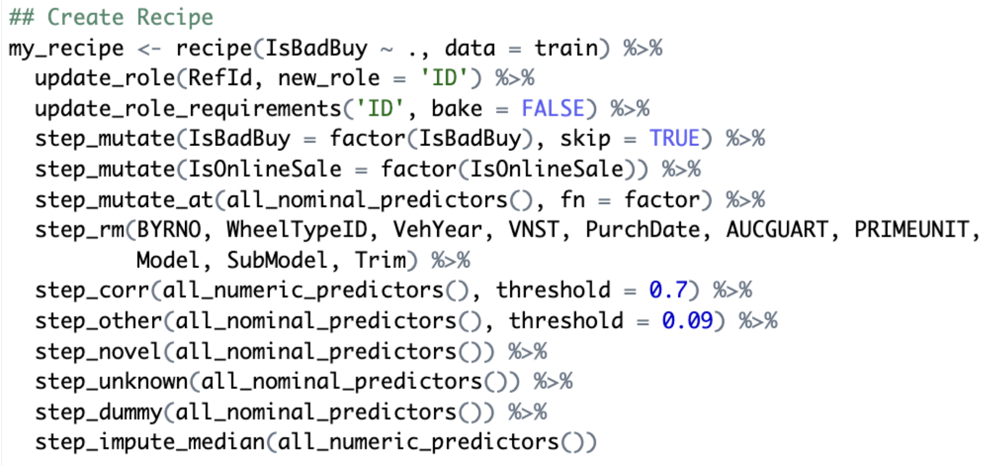
FINAL KAGGLE COMPETITION: DON’T GET KICKED

By: Angelina Camen

Problem Background

- Auto dealerships face significant financial risk when purchasing used cars at auctions. Some vehicles arrive with hidden issues such as odometer tampering, unfixable mechanical problems, or title complications that prevent them from being sold.
- These costly problem vehicles are known as **“kicks.”**
- Because kicks lead to transportation losses, wasted repair costs, and reduced resale value, dealerships benefit greatly from identifying high-risk vehicles before buying them.
- GOAL: To predict whether a purchased auction vehicle will be a “bad buy,” helping dealerships make smarter and safer inventory decisions.



```r
## Create Recipe
my_recipe <- recipe(IsBadBuy ~ ., data = train) %>%
  update_role(RefId, new_role = 'ID') %>%
  update_role_requirements('ID', bake = FALSE) %>%
  step_mutate(IsBadBuy = factor(IsBadBuy), skip = TRUE) %>%
  step_mutate(IsOnlineSale = factor(IsOnlineSale)) %>%
  step_mutate_at(all_nominal_predictors(), fn = factor) %>%
  step_rm(BYRNO, WheelTypeID, VehYear, VNST, PurchDate, AUCGUART, PRIMEUNIT,
          Model, SubModel, Trim) %>%
  step_corr(all_numeric_predictors(), threshold = 0.7) %>%
  step_other(all_nominal_predictors(), threshold = 0.09) %>%
  step_novel(all_nominal_predictors()) %>%
  step_unknown(all_nominal_predictors()) %>%
  step_dummy(all_nominal_predictors()) %>%
  step_impute_median(all_numeric_predictors())
```

Description of Feature Engineering

- **Identify ID Column**

- *RefId* is marked as an identifier so it is not used as a predictor in the model.

- **Convert Key Variables to Factors**

- Important categorical fields such as *IsOnlineSale* and all other nominal predictors are converted to factors so the model treats them as categories instead of numbers.

- **Remove Low-Value or Redundant Columns**

- Variables that offered little predictive power or introduced noise—such as *VehYear*, *Model*, *Trim*, and others—were removed to simplify the model and improve performance.

- **Handle Highly Correlated Numeric Predictors**

- Numeric predictors with a correlation above 0.7 were removed to reduce multicollinearity and prevent redundant information from confusing the model.

- **Combine Rare Categories**

- Rare factor levels (threshold = 0.09) were grouped into an “Other” category to improve model stability.

- **Prepare for Unseen Categories**

- step_novel() and step_unknown() ensure the model can handle new or missing category values that appear in the test set.

- **Convert Categorical Variables to Dummy Variables**

- One-hot encoding (step_dummy()) transforms all factor variables into numeric indicator columns required by many machine learning models.

- **Impute Missing Numeric Values**

- Missing values in numeric predictors are filled using median imputation, a robust method that avoids distortion from outliers.

Model Comparison

Model 1: Random Forest + LightGBM Stacked Ensemble	Model 2: BART (Bayesian Additive Regression Trees)	Model 3: XGBoost (Boosted Trees)
• Combined two powerful learners (Random Forest + LightGBM) using stacking. • Used cross-validation and random grid search for efficient tuning. • Produced good overall performance, but slightly below the target. • **Submission Score: 0.23216**	• Used BART with tuned tree counts to capture nonlinear structure in the data. • Included a simplified but carefully prepared recipe with dummy variables, correlation filtering, and median imputation. • BART's additive tree structure handled complex interactions particularly well for this dataset. • **Leaderboard Score: 0.23576**	• Gradient boosting model tuned with Latin hypercube sampling (trees, learn_rate, mtry, tree_depth) • Fast to train and highly efficient with large datasets • Automatically handles nonlinear relationships and variable interactions • Works especially well after recipe preprocessing (dummy variables, imputation, correlation filtering) • **Leaderboard Score: 0.17332**

Details of Best Model: BART

- The BART model works by combining *many* small regression trees, each contributing a tiny part to the final prediction. This "additive" structure helps it capture subtle, nonlinear relationships in the vehicle features.
- Unlike traditional tree models, BART uses Bayesian principles, meaning it places priors on the tree structures and automatically regularizes the model. This prevents overfitting, which is especially important in a noisy dataset like *Don't Get Kicked*.
- BART excels in situations where interactions between variables are hard to hand-engineer. It can naturally uncover complex patterns related to things like mileage, age of vehicle, purchase auction, or condition.
- The model produces probabilistic predictions rather than just fixed outputs. This gives more stable classification behavior by smoothing overly confident predictions from individual trees.
- Because BART tends to emphasize generalizable structure rather than deep, highly variable trees, it handled the messy, categorical-heavy nature of the dataset more reliably than other models tested.