Appendix E: Parametric Model vs. ML (Decision Tree)

Prior to addressing the main research objective, we demonstrate that even a simple ML approach, a regression tree, can achieve comparable results with parametric models.[1] The SKU dataset is split into a training set (80%) with 7,457 observations and a test set (20%) with 1,864 observations. Building the regression tree requires the selection of a complexity parameter (CP) resulting in the minimum root mean square error (RMSE). Training and test error estimation follows the parameter tuning by looping over 100 random data splits to identify the CP with minimum test error. CP imposes a penalty to the tree for having too many splits. The higher the CP, the smaller the tree. A too-small CP value leads to overfitting and a too-large CP value will result in underfitting.

An optimal CP value can be estimated by testing different CP values and using cross-validation. The optimal CP is defined as the CP that minimizes the cross-validation error (RMSE). Ultimately the CP determines the tree's number of leaves and branches in the attempt to find the most parsimonious tree that gives the best fit (i.e., pruning). The regression tree needs pruning to avoid any overfitting of the data based on the CP with the objective of reducing the complexity of the final classifier. We finally visualize the tree, plot the bins according to the split values, and report the averaged training and test errors.

[1] We use the Caret package in R for the ML approach (Kuhn 2008). For data splitting we use the createDataPartition function in R's Caret package to form random stratified splits of the data (Kuhn 2020) while maintaining balance (i.e., the ratio of classes is maintained in the splits). Given the imbalance of classes, this approach avoids the potential problem of having too few minority class examples in the training and test data (Batista, Prati and Monard 2004).

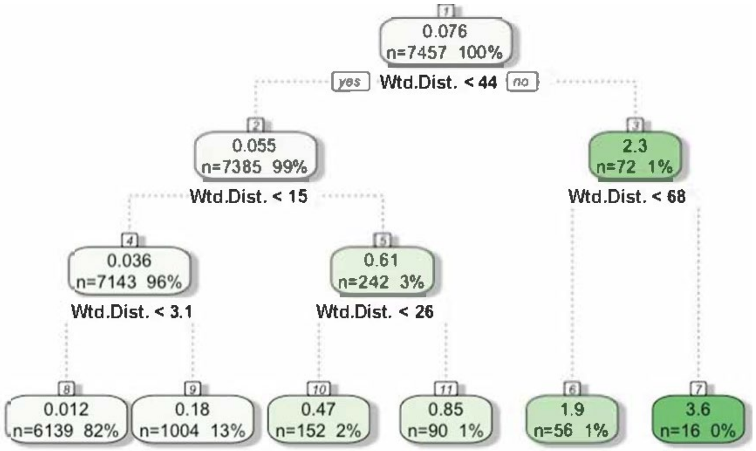

Fig. E1. Simple decision tree estimating market share based on distribution.

Figure E1 presents the final tree with the split criterion (e.g., Wtd. Dist.<44), the number of observations in that branch (n), and the market share estimate for the branch (averaged value of observations in this branch). Although a regression tree can be displayed graphically and is easy to explain, it suffers from high variance, which can result in high estimation error on unseen data. By binning the market share estimates and the equivalent threshold values of distribution, we observe the expected convex pattern in the relationship (see Figure E2).

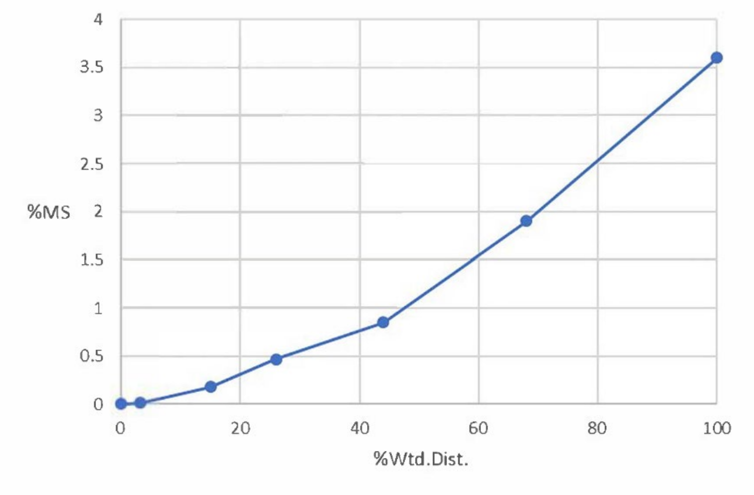

Fig. E2. Regression tree bins projected as velocity curve.

The regression tree produces similar training and test error, indicating good fit. The average R-squared is 63% compared with 72% in the NLS. The ML approach makes equivalent estimations for the distribution – market share relationship, offering an initial validation of the ML approach. The NLS slightly outperforms the regression tree, suggesting the simple boundary of linear or curvilinear regression models generalizes slightly better.