Predictive Models Group Project: Pricing Cars

Mark Babbe, Camryn Callaway, Siqi Chen, Korawat Tanwisuth, Zhiyi Yang

Introduction

Our goal is to construct a predictive model in order to make predictions on car prices based on given features. To create and validate our models, we randomly split the dataset into training and test sets, each containing 70% and 30% of the data respectively. We use the training set to fit proposed models and use the test set to validate the performance of each model regarding prediction accuracy. For each model, we use cross validation within the training set to select the parameters that give the lowest cross-validated RMSE. In the end, we make comparison among all models and select the one with the best prediction accuracy. The following three categories of models are proposed: linear regression models, K-nearest neighbor, tree-based models.

Exploratory Data Analysis

We first examine the data to see if there is any missing value or unrelated variables. The dataset `Cars.csv` contains 17 variables and 29466 observations. We notice that variable `X` is simply the index of each observation which is unrelated for predicting `price`, hence it's removed from the dataset. Out of the remaining 15 variables other than the response variable `price`, 3 are numerical and 12 are categorical. For numerical predictors `mileage` and `year`, scatter plots versus `price` show that there might be some polynomial relationship, which could be further examined in linear regression models. For categorical predictors, there exist many observations with the factor level `unsp`, which we initially believe that we could treat them as missing values. However, box plots for each of the categorical predictors versus price indicate that `unsp` itself as a category could be informative for `price`, so we keep it as a factor level.

We also find two pairs of perfectly correlated predictors: (`subTrim`, `fuel`), (`state`, `region`). There're only two levels in `subTrim`: `hybrid` and `unsp`, while `hybrid` is already included in `fuel` as a fuel type. Also, `region` can be expressed as a linear combination of `state` since each state corresponds to a specific region. However, `region` is defined based on employment status, population composition, weather conditions, etc. which could be more informative than `state`. Furthermore, some levels in `state` contains only few data points, which could result in biased prediction. Therefore, we decide to remove `subTrim` and `state` from the predictors.

Linear Regression Models

We first fit a multiple linear regression model with all 14 predictors and examine the residual plot for necessary variable transformation. The plot shows that there's an increasing trend in residual for higher `price` values. Thus we perform a log transformation on `price`. The resulting residual plot shows a relatively random pattern, with significant reduction in $RMSE^o$. We then use stepwise regression to examine the significance of $\texttt{mileage}^2$ and $\texttt{year}^2$ as suggested in EDA. Both results from forward and backward selection include the two second order terms which indicate the significance. The next step is to examine the potential two-way interactions, since the predictors consist a large portion of categorical variables. The interaction plots of following

nine pairs of predictors indicate significant interactions: (`mileage`,`condition`), (`mileage`,`trim`), (`mileage`,`displacements`), (`year`,`condition`), (`year`,`displacement`), (`year`,`trim`), (`condition`,`region`), (`condition`,`trim`), (`condition`,`color`).

This led us to utilize shrinkage methods to reduce variance in coefficients, which ultimately reduce $RMSE^o$. We fit all the predictors including second order terms and interactions in ridge and lasso regression, and perform 10-fold cross validation within training set to determine the best α. The regression results shows significant reduction in $RMSE^o$ comparing to multiple linear regression model, and lasso outperforms ridge with slightly better prediction accuracy.

Since the dimension of predictors in this dataset is relatively large, we also try to derive a lower-dimensional set of features using principle component analysis. We fit the principle components to partial least square regression, and the resulting $RMSE^o$ is slightly higher than ridge and lasso regression.

K-Nearest Neighbors

In general, KNN regression performs poorly comparing to linear regression in high dimensions, as a result of curse of dimensionality. In order to examine the actual performance, we first standardize 3 numerical predictors to reduce bias. We then perform 5-fold and 10-fold cross validation to determine the best K. Cross-validated $RMSE$ from both models suggest $K = 6$ for the lowest prediction error. The resulting $RMSE^o$ from 6-nearest neighbors regression is much higher than linear regression models as expected.

Tree-Based Models

Tree-based models captures potential non-linearity and interactions, thus we fit all 14 predictors to three tree-based models: single tree, random forest and boosting. We anticipate that they will outperform the models we have attempted so far as tree-based models are more perceptive to non-linear relationships. To no surprise, $RMSE^o$ is further reduced in all three models.

For single tree model, we perform 10-fold cross validation to determine the best tree size (number of terminal nodes) for the lowest $RMSE$. For random forest model, we perform 5-fold cross validation to determine the best combination of number of predictors (`mtry`) and number of trees (`n.trees`). For boosting model, we also perform 5-fold cross validation to determine the best combination of number of split (`interaction.depth`), number of trees (`n.trees`) and λ (`shrinkage`). As expected, $RMSE^o$ from single tree model is slightly higher than random forest and boosting models, since the latter two models reduces bias through resampling and stepwise residual-fitting respectively.

Conclusion

Our final model selection narrows down to random forest and boosting models. $RMSE^o$ from both models are very close in several training and test sets samplings, so we decide to compare complexity of two models to prevent potential overfitting. Best tunes for random forest are `mtry` $= 14$ and `n.trees` $= 200$, whereas for boosting are `interaction.depth` $= 16$, `n.trees` $= 2000$ and `shrinkage` $= 0.01$. Therefore, we choose random forest as our final model for predicting car prices. `mileage`, `year`, `condition`, `trim` and `wheelSize` are the top 5 "important" predictors regarding the reduction in loss due to the splits using these predictors, which are also intuitively important when we price a car.