Areknaz Khaligian
May 23, 2019

Advanced R
Individual Project
Predicting Black Friday Purchase

The objective of this project is to predict Black Friday purchases based on "BlackFriday_train.csv" to generate predictions for "BlackFriday_test.csv".

The general steps are outlined below:

1. Feature Engineering
2. Model Training
3. Model Tuning
4. Generating Predictions

In the feature engineering section we notice some key points:

(a) All of the features can be interpreted as factor variable (other than the target variable: Purchase, which is numeric)
(b) The only feature with missing values are those with not applicable Product Category 2 or 3 (will need to decide how to treat these NAs)
(c) Some factors have levels with less than 5% of the total number of observations
(d) There are some features related to customer or product ID which should be discarded, since the model should be based on product/customer characteristics not unique ID values.

At this point there are two remaining steps of data cleaning before proceeding to Model Training: treating the missing values in Product Category 2 and Product Category 3 as well as grouping the factor levels with less than 5% of the total observations. As a baseline, replace missing Product Category 2's with the respective Product Category 1, and the missing Product Category 3's with the respective Product Category 2. For the factor levels, any factor level with less than 5% total observations will be grouped as an "Others" level.

Here are the results of the Model Training for the baseline model:

0 Baseline

	Method	rmse	mae	mape	
1:	tree	3990.531	3032.263	0.6601373	*general tree based model*
2:	ctree	3840.119	2867.627	0.6205934	*tree with conditional probabilities*
3:	rf1	3834.422	2873.827	0.6359518	*random forest model*
4:	xgb	3819.784	2854.733	0.6191970	*xgboost model*
5:	lm	4012.684	3031.127	0.6754721	*linear model (stepwise selection)*
6:	glmnet	4012.678	3030.877	0.6756331	generalized linear model
7:	xgb_reg	4032.222	3061.196	0.6765542	*xgboost regression*

The data was run through 7 different models. In yellow are the top 4 performing models, with green highlighting the best model in terms of MAPE. Going forward only these 4

models will be run since they show the highest potential for the tree or basic regression approaches.

In the next step create a Version 1 for data cleaning/engineering. Missing values in Product Category 2 / 3 are treated the same as before, but for the factor levels less than 5% instead of blindly grouping into Other categories, divide into “high purchase other” and “low purchase other” categories based on the median purchase of the entire dataset. For example, if Product Category = 10 has less than 5% of observations, and the median purchase value for Product Category = 10 is above the median purchase value for the entire dataset, these values will be recoded as “high”.

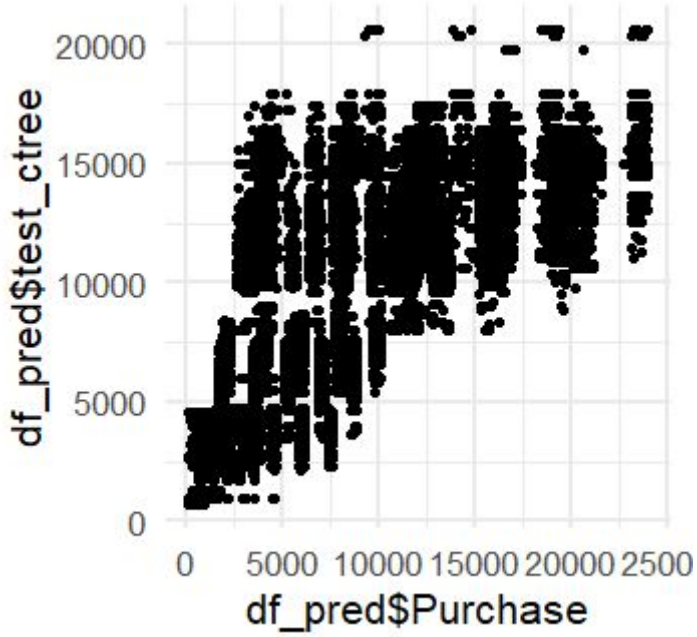
Here are the results of the Model Training for the version 1 model:

1 High/Low other categories

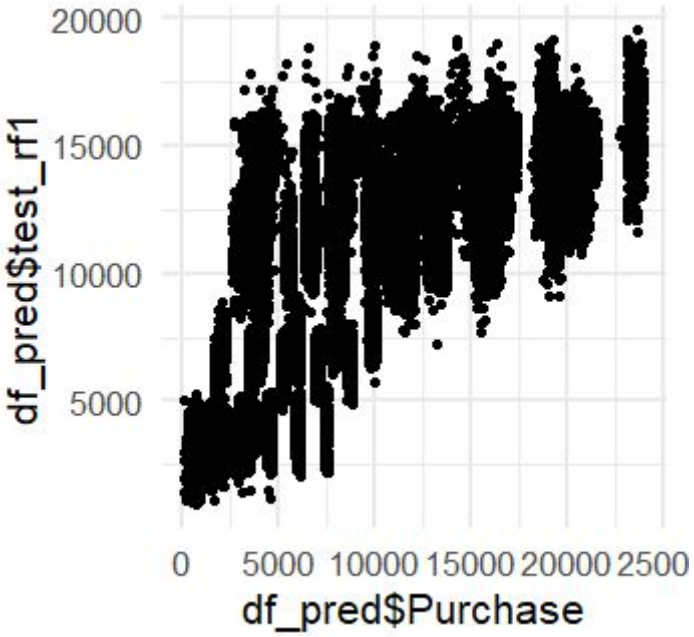
	Method	rmse	mae	mape	
1:	ctree	3101.626	2341.702	0.3827651	*tree with conditional probabilities*
2:	rf1	3077.969	2326.782	0.3883433	*random forest model*
3:	xgb	3082.843	2332.146	0.3823783	*xgboost model*
4:	lm	3219.059	2450.091	0.4118041	*linear model (stepwise selection)*

Here xgb is still the best performing model, but with the change in feature engineering all of the MAPE scores are greatly improved. (See appendix for other versions which did not perform as well).

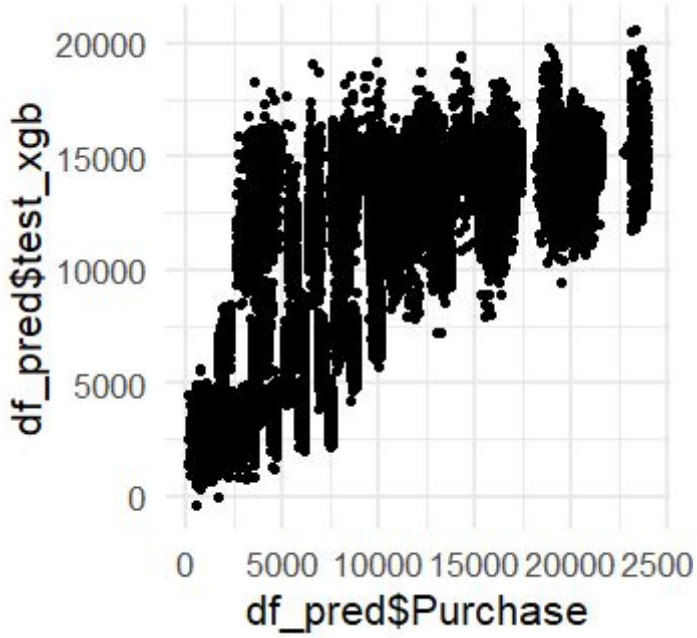
Here are the results of plotting the predicted (df_pred$test_model) vs real values (df_pred$Purchase) for each model:



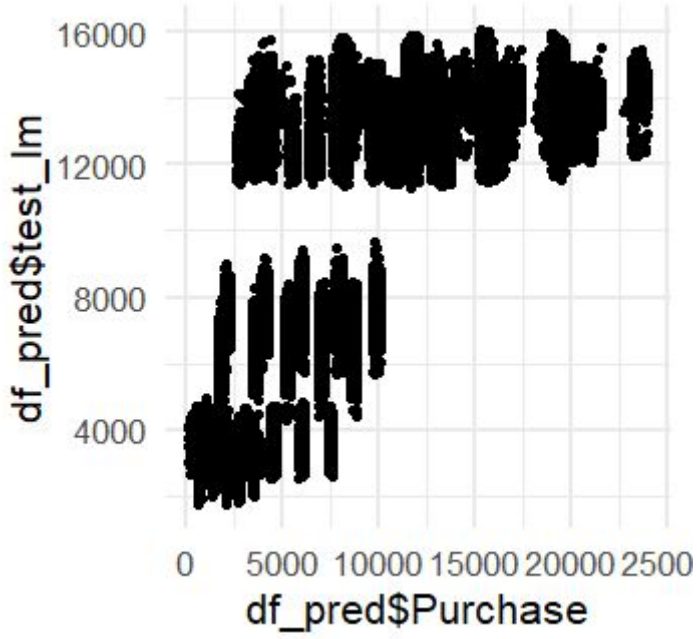
This model’s predictions seem to be regularly underestimating the real values especially as the values increase.



This model also results in underestimating the real value, but the predictions are more tightly clumped together.



This model is also similar to the previous tree based models, tending to underestimate the real value, but clumping even more tightly together (this model had the best MAPE score)



This linear based model is clearly performing worse than the tree based models. The underestimation is even worse and there are awkward groups of values which do not follow the trends of the true values.

In the next step we will optimize the parameters of each model to create the final model. Below are the results of the optimization, comparing the train and test errors:

PARAMETER OPTIMIZATION

	Method	mape-train	mape-test
1:	ctree	0.3792264	0.3819125
2:	rf1	0.3582362	0.378867
3:	xgb	0.3815775	0.3840332
4:	lm	NA	NA

It was clear that the linear model did not have the capabilities to compete with the tree based models (and a stepwise selection linear model does not have any tuning parameters) so it was excluded from the options to choose our final model. After optimizing the parameters, the random forest model was able to best capture the data with the best MAPE score. One noteworthy point is that the difference between the test and train score was slightly higher with the random forest model (even though the final test score was better) compared to the other models. This could suggest that the random forest model is not generalizing quite as well as ctree and xgb models, but in the end the difference is not enough to justify not selecting the rf model as the final choice.

Appendix

---

2 Adding interactions between features

Since tree based models are able to capture interactions, this is only relevant for the linear based model. Unfortunately the machine I was working on did not have the RAM capacity to run the linear model with all feature interactions even with a smaller sample size. This was justified since research shows that tree based models are able to more efficiently capture interactions than linear models regardless of the regression formula.

In this version 3 of the data cleaning/engineering, another approach to the missing values was considered. Instead of re-coding the Product Category Codes for other known codes, they were set to 0. One issue with this was that it affected the grouping of low frequency factor levels into high and low values. As a result, it performed poorly and was not selected as a good data cleaning/engineering approach.

3 NA Product Categories to 0

	Method	rmse	mae	mape
1:	ctree	3921.317	2934.860	0.6308149
2:	rf1	3900.069	2918.952	0.6391261
3:	xgb	3893.826	2909.421	0.6267253
4:	lm	4040.085	3053.653	0.6758181