Module 6 HW

Alanna Hazlett

2024-06-24

Module 6

> You can download the R Markdown file (https://gedeck.github.io/DS-6030/homework/Module-6.Rmd (https://gedeck.github.io/DS-6030/homework/Module-6.Rmd)) and use it to answer the following questions.
>
> If not otherwise stated, use Tidyverse and Tidymodels for the assignments.

1. Predict out of state tuition (feature selection)

The data College.csv contains a number of variables for 777 different universities and colleges in the US. In this exercise, we will try to predict the Outstate tuition fee using the other variables in the data set.

(a.) Load the data from ISLR2::College and split into training and holdout sets using a 80/20 split.

```
college <- ISLR2::College
##??college
#Private is already factored
set.seed(1)  # for reproducibility
college_split <- initial_split(college, prop=0.80, strata=Outstate)
college_train <- training(college_split)
college_test <- testing(college_split)
#summary(college)
```

```
college_formula<-Outstate~Private+Apps+Accept+Enroll+Top10perc+Top25perc+`F.Undergrad`+`P.Undergrad`+`Room.Board`
+Books+Personal+PhD+Terminal+`S.F.Ratio`+`perc.alumni`+Expend+`Grad.Rate`

lm_recipe<-recipe(college_formula, data=college) %>%
      step_normalize(all_numeric_predictors()) %>%
      step_dummy(all_nominal_predictors())
```

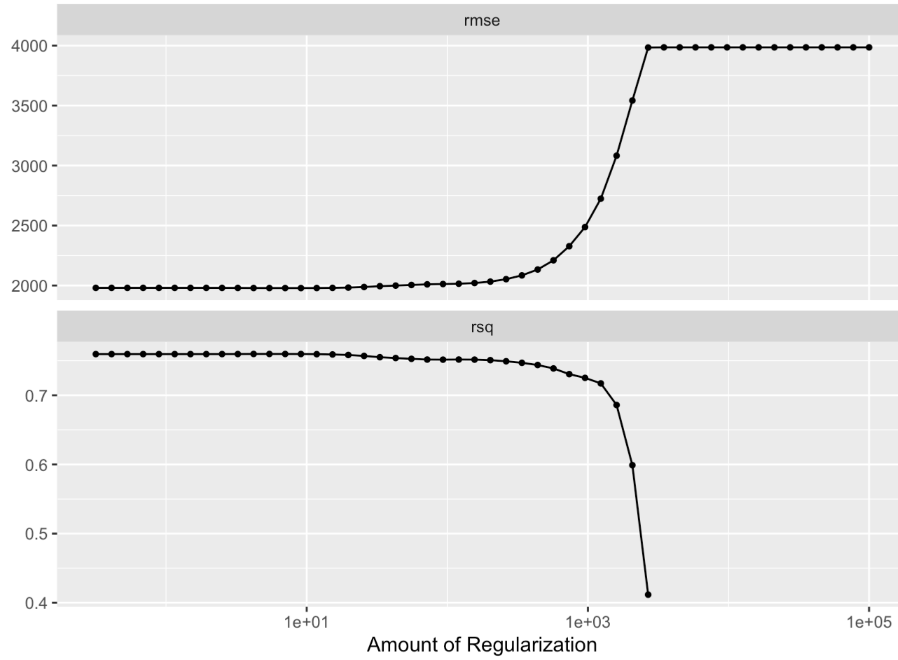
(ii.) Use glmnet and tune the L1 penalty parameter using 10-fold cross-validation. Make sure you select an appropriate range for the tuning parameter.

```
#L1 Lasso is mixture value of 1
tune_lm_spec<-linear_reg(engine="glmnet", mode="regression", mixture=1, penalty=tune())

tune_lm_wf<- workflow() %>%
    add_model(tune_lm_spec) %>%
    add_recipe(lm_recipe)

#Assumes penalty is between [-10,0]
#Try wide range for penalty, find variance, and narrow range to that area
lm_params <- extract_parameter_set_dials(tune_lm_wf) %>%
  update(penalty=penalty(c(-0.5, 5)))
```

Cross-validation combined with tuning

```
resamples<-vfold_cv(college_train,v=10,strata=Outstate)
cv_control <- control_resamples(save_pred=TRUE)
custom_metrics<-metric_set(rmse,mae,rsq)

tune_results_lm <- tune_grid(tune_lm_wf,
                          resamples=resamples,
                          control=cv_control,
                          grid=grid_regular(lm_params, levels=50))
```

```
autoplot(tune_results_lm)
```



Model performance across levels of penalty.

(iii.) Select the best model based on RMSE and report the coefficients of the model. Which variables are selected by the model? **(iv.)** Train a finalized model using the best tuning parameter and report the RMSE and R Squared of the model on the training and test set. What do you observe?

> *Do to order of code these two parts are mixed together in this section*

```
#Finding best model

#n is number of top models we want
show_best(tune_results_lm, metric='rmse', n=1) %>%
    knitr::kable(digits=3,caption="Table 1: Best Model") %>%
    kableExtra::kable_styling(full_width=FALSE)
```

Table 1: Best Model

penalty	.metric	.estimator	mean	n	std_err	.config
9.103	rmse	standard	1978.853	10	67.919	Preprocessor1_Model14

```
#Fitting best model and calculating metrics
lm_final<-finalize_workflow(tune_lm_wf, select_best(tune_results_lm, metric='rmse')) %>%
                                  fit(college_train)
lm_metrics<-bind_rows(
            bind_cols(Dataset="LM Train",metrics(augment(lm_final,college_train),truth=Outstate,estimate=.pred)),
            bind_cols(Dataset="LM Test",metrics(augment(lm_final,college_test),truth=Outstate,estimate=.pred)))
lm_tab<-lm_metrics %>%
    pivot_wider(id_cols=Dataset, names_from=.metric, values_from=.estimate)

lm_tab%>%
    knitr::kable(digits=3,caption="Table 2: Best Linear Model Metrics") %>%
    kableExtra::kable_styling(full_width=FALSE)
```

Table 2: Best Linear Model Metrics

Dataset	rmse	rsq	mae
LM Train	1915.129	0.769	1524.103
LM Test	2071.280	0.751	1576.511

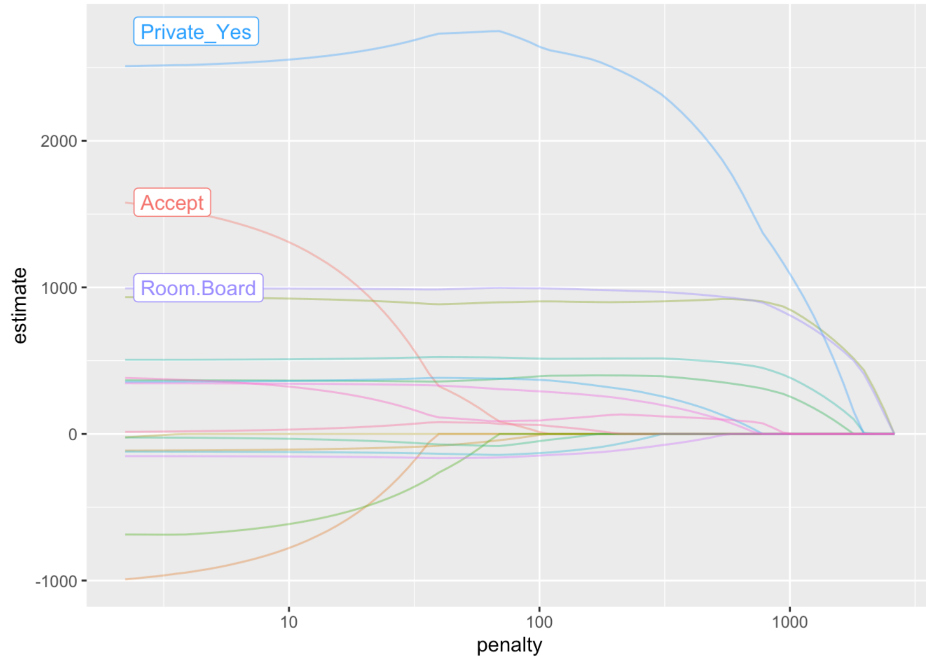
> In Table 2 we see that as we would expect the model performed slightly better on the training data than it did on the new test data, with lower rmse and mae values and a higher R squared value for the training data.

```
#Coefficients of best model
tidy(lm_final) %>%
  #filter(term != '(Intercept)')
  knitr::kable(digits=3,caption="Table 3: Coefficients of Best Model") %>%
  kableExtra::kable_styling(full_width=FALSE)
```

Table 3: Coefficients of Best Model

term	estimate	penalty
(Intercept)	8583.846	9.103
Apps	-802.763	9.103
Accept	1339.140	9.103
Enroll	0.000	9.103
Top10perc	329.643	9.103
Top25perc	28.093	9.103
F.Undergrad	-625.218	9.103
P.Undergrad	-31.987	9.103
Room.Board	992.390	9.103
Books	-107.446	9.103
Personal	-122.620	9.103
PhD	361.070	9.103
Terminal	343.431	9.103
S.F.Ratio	-153.426	9.103
perc.alumni	508.965	9.103
Expend	924.695	9.103
Grad.Rate	364.805	9.103
Private_Yes	2548.316	9.103

```
autoplot(lm_final %>%  extract_fit_engine())
```



Coefficient values accross different penalty values

> In Figure 2 we see that our selected penalty that minimizes the rmse of 9.103 nearly all of the coefficients still have values that are not zero. The only coefficient that is zero at this level of penalty is Enroll.

(c.) Using the selected features from (b), build a generalized additive model (GAM) to predict Outstate. (see Generalized additive models (GAM) (https://gedeck.github.io/DS-6030/book/deep-dive-gen_additive_mod.html) for how to build GAM models in tidymodels) **(i.)** Define a model formula setting all numerical variables as splines and all categorical variables as factors (Outstate ~ Private + s(Apps) + …)

```
gam_formula<-Outstate~Private+s(Apps)+s(Accept)+s(Top10perc)+s(Top25perc)+s(`F.Undergrad`)+s(`P.Undergrad`)+s(`Ro
om.Board`)+s(Books)+s(Personal)+s(PhD)+s(Terminal)+s(`S.F.Ratio`)+s(`perc.alumni`)+s(Expend)+s(`Grad.Rate`)
```

(ii.) Define the gen_additive_mod model using mgcv as the engine and fit the model using the training data.

```
gam_model <- gen_additive_mod() %>%
    set_engine("mgcv") %>%
    set_mode("regression") %>%
    fit(gam_formula, data=college_train)
```

(iii.) Report the RMSE and R Squared of the model on the training and test set. What do you observe? How does it compare to (b.iv)?

```
gam_metrics<-bind_rows(
            bind_cols(Dataset="GAM Train",metrics(augment(gam_model,college_train),truth=Outstate,estimate=.pre
d)),
            bind_cols(Dataset="GAM Test",metrics(augment(gam_model,college_test),truth=Outstate,estimate=.pred)))
gam_tab<-gam_metrics %>%
    pivot_wider(id_cols=Dataset, names_from=.metric, values_from=.estimate)

knitr::kables(list(knitr::kable(lm_tab,digits=2) %>%  kableExtra::kable_styling(full_width=FALSE),
                 knitr::kable(gam_tab,digits=2) %>%  kableExtra::kable_styling(full_width=FALSE)), caption="Metri
cs for LM and GAM")
```

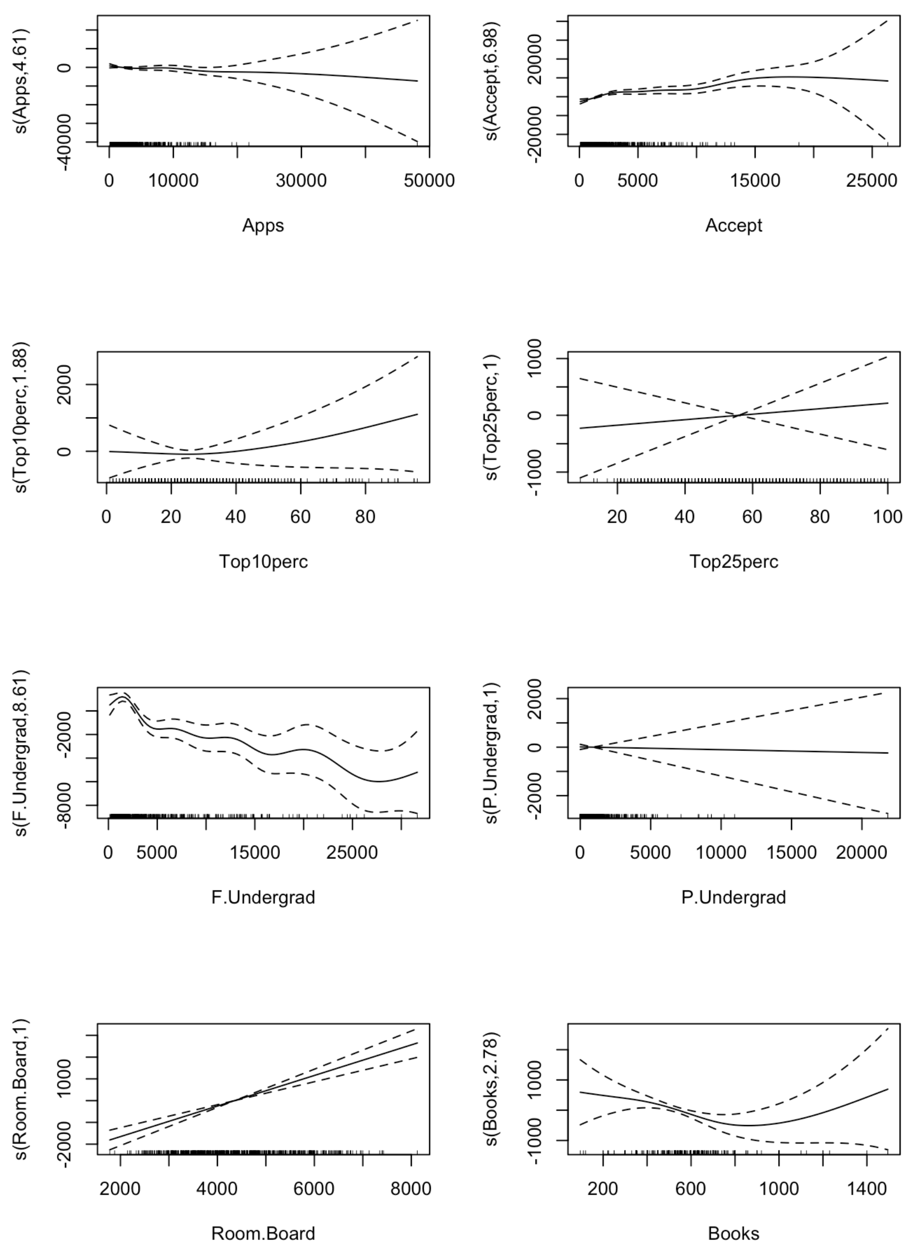
Metrics for LM and GAM

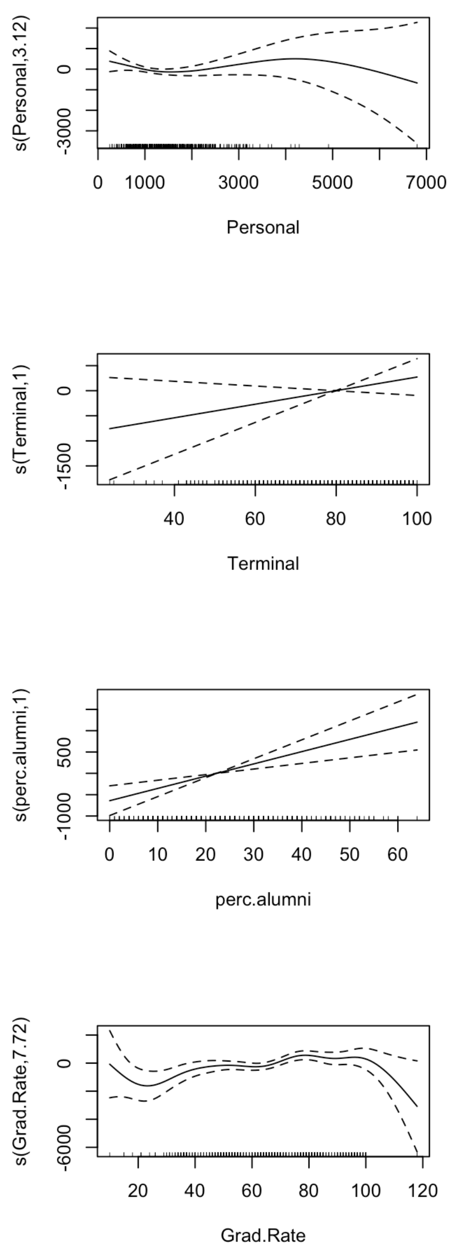
Dataset	rmse	rsq	mae	Dataset	rmse	rsq	mae
LM Train	1915.13	0.77	1524.10	GAM Train	1591.30	0.84	1256.23
LM Test	2071.28	0.75	1576.51	GAM Test	1923.45	0.79	1434.37

> We do see improvement in the GAM with splines compared the the LM with L1 regularization. The GAM has lower values for rmse and mae and a higher R squared. As we would expect the test values for GAM are slightly worse than it’s training values.

(iv.) Use the plot with the fitted model (use extract_fit_engine to get the actual mgcv model for plotting). Describe your observations.

```
opar <- par(mfrow=c(2,2))
plot(gam_model %>% extract_fit_engine(), scale=0)
```





> The GAM model fits spline with for Apps, Accept, Top10perc, F.Undergrad, P.Undergrad, Books, Personal, Terminal, S.F.Ratio, Expend, and Grad.Rate with considerable non-linearity. This is notable based on the uneven distribution of the observations seen in the rug plots for each graph.

(v.) Use the summary function to get information about the model. Based on the reported significance levels, could you simplify the model further?

```
tidy(gam_model %>% extract_fit_engine()) %>%
  knitr::kable(digits=3,caption="Table 5: Coefficients of GAM with Splines") %>%
  kableExtra::kable_styling(full_width=FALSE)
```

Table 5: Coefficients of GAM with Splines

term	edf	ref.df	statistic	p.value
s(Apps)	4.614	5.558	1.580	0.242
s(Accept)	6.982	7.772	4.623	0.000

term	edf	ref.df	statistic	p.value
s(Top10perc)	1.884	2.438	1.231	0.338
s(Top25perc)	1.000	1.000	0.272	0.602
s(F.Undergrad)	8.615	8.947	5.369	0.000
s(P.Undergrad)	1.000	1.000	0.037	0.848
s(Room.Board)	1.000	1.000	64.297	0.000
s(Books)	2.783	3.551	3.223	0.016
s(Personal)	3.118	3.929	1.407	0.238
s(PhD)	1.543	1.928	0.815	0.382
s(Terminal)	1.000	1.000	2.203	0.138
s(S.F.Ratio)	4.840	5.964	2.933	0.008
s(perc.alumni)	1.000	1.000	13.554	0.000
s(Expend)	6.421	7.564	14.254	0.000
s(Grad.Rate)	7.723	8.546	2.816	0.003

> We see high p-values for s(Apps),s(Top10perc),s(Top25perc),s(P.Undergrad),s(Personal),s(PhD),s(Terminal),s(S.F.Ratio).

(vi.) Simplify the model by removing the non-significant variables and re-fit the model. Report the RMSE and R Squared of the model on the training and test set. What do you observe?

```
gam_reduced_formula<-Outstate~Private+s(Accept)+s(`F.Undergrad`)+s(`Room.Board`)+s(Books)+s(`perc.alumni`)+s(Expe
nd)+s(`Grad.Rate`)

gam_reduced_model <- gen_additive_mod() %>%
    set_engine("mgcv") %>%
    set_mode("regression") %>%
    fit(gam_reduced_formula, data=college_train)

gam_reduced_metrics<-bind_rows(
            bind_cols(Dataset="Reduced GAM Train",metrics(augment(gam_reduced_model,college_train),truth=Outstat
e,estimate=.pred)),
            bind_cols(Dataset="Reduced GAM Test",metrics(augment(gam_reduced_model,college_test),truth=Outstate,e
stimate=.pred)))
gam_reduced_tab<-gam_reduced_metrics %>%
    pivot_wider(id_cols=Dataset, names_from=.metric, values_from=.estimate)
gam_reduced_tab %>%
    knitr::kable(digits=3,caption="Table 6: Reduced GAM Metrics") %>%
    kableExtra::kable_styling(full_width=FALSE)
```

Table 6: Reduced GAM Metrics

Dataset	rmse	rsq	mae
Reduced GAM Train	1669.384	0.825	1317.669
Reduced GAM Test	1895.214	0.793	1451.475

(d.) Compare the results from the three models (b) and (c)?

```
knitr::kables(list(knitr::kable(lm_tab,digits=2) %>%  kableExtra::kable_styling(full_width=FALSE),
                  knitr::kable(gam_tab,digits=2) %>%  kableExtra::kable_styling(full_width=FALSE),
                  knitr::kable(gam_reduced_tab,digits=2) %>%  kableExtra::kable_styling(full_width=FALSE)),captio
n="Metrics for LM and GAM")
```

Metrics for LM and GAM

Dataset	rmse	rsq	mae	Dataset	rmse	rsq	mae	Dataset	rmse	rsq	mae
LM Train	1915.13	0.77	1524.10	GAM Train	1591.30	0.84	1256.23	Reduced GAM Train	1669.38	0.82	1317.67
LM Test	2071.28	0.75	1576.51	GAM Test	1923.45	0.79	1434.37	Reduced GAM Test	1895.21	0.79	1451.48

> In Table 7 we see kind of mixed results from our Reduced GAM. The rmse for the test data is lower for rmse compared to LM and full GAM, indicating some improvement, but the R squared is the same value for the reduced GAM as the full GAM and the mae is actually higher for the reduced GAM compared to the full GAM on the test data.

Stop cluster

```
stopCluster(cl)
registerDoSEQ()
```