Regression Methods on Prostate Cancer Data

Shreyas Srinivasan

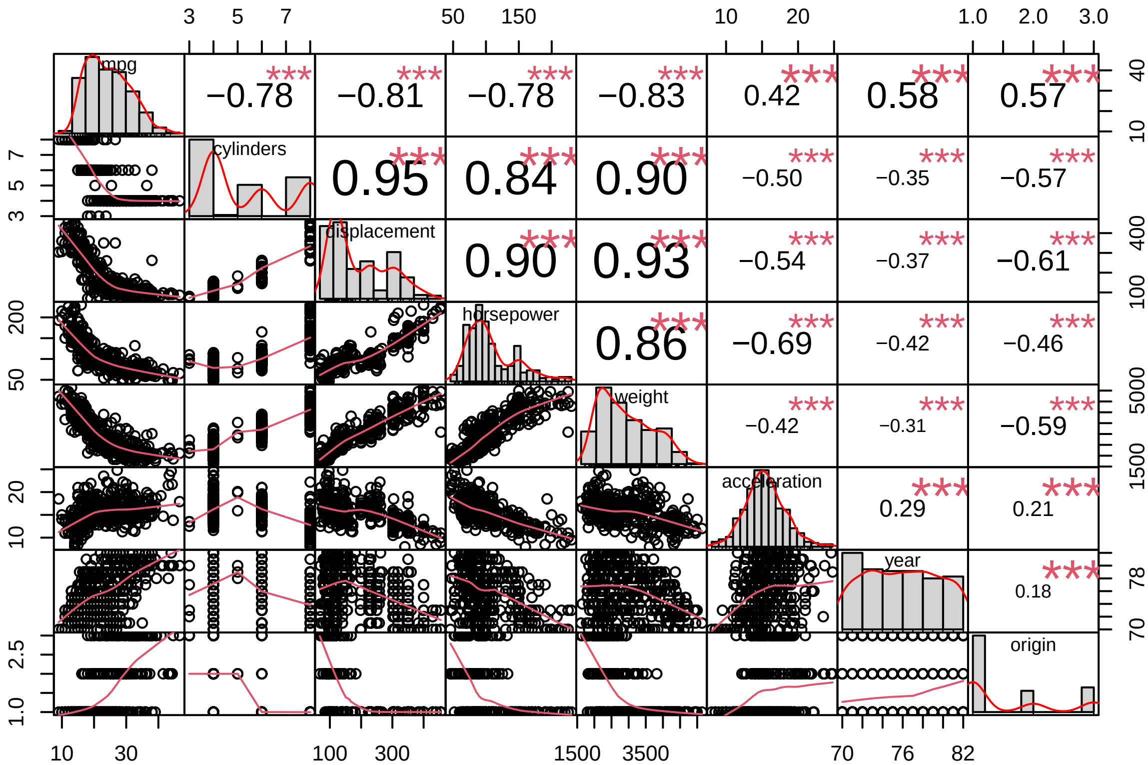
Dataset

The `Auto` dataset is available in the `ISLR` package. The dataset contains 392 observations with 9 attributes for each observation. The attributes are briefly described below:

1. mpg - miles per gallon
2. cylinders - Number of cylinders between 4 and 8
3. displacement - Engine displacement (cu. inches)
4. horsepower - Engine horsepower
5. weight - Vehicle weight (lbs.)
6. acceleration - Time to accelerate from 0 to 60 mph (sec.)
7. year - Model year (modulo 100)
8. origin - Origin of car (1. American, 2. European, 3. Japanese)
9. name - Vehicle name

Our goal is to build a model that can predict `mpg`. We want to be able to predict the mileage of a vehicle from other attributes.

```
#exploratory analysis
chart.Correlation(Auto[, -9])
```



From the graph above, we see that a bunch of predictors are highly correlated with each other. For example, weight and displacement have a correlation coefficient of 0.93. This suggests that 1 (or more) predictors may not be useful in predicting mpg. When we look at the relationship between the response (mpg) and other variables, acceleration does not show a strong relationship with mpg. Every other variable has a correlation coefficient > 0.50 with mpg.

We will consider the following regression methods:

(a) Standard Least Squares

(b) Best-subset selection

(c) Ridge regression

(d) Lasso regularization

(e) Principal Component Regression (PCR)

(f) Partial Least Squares (PLS)

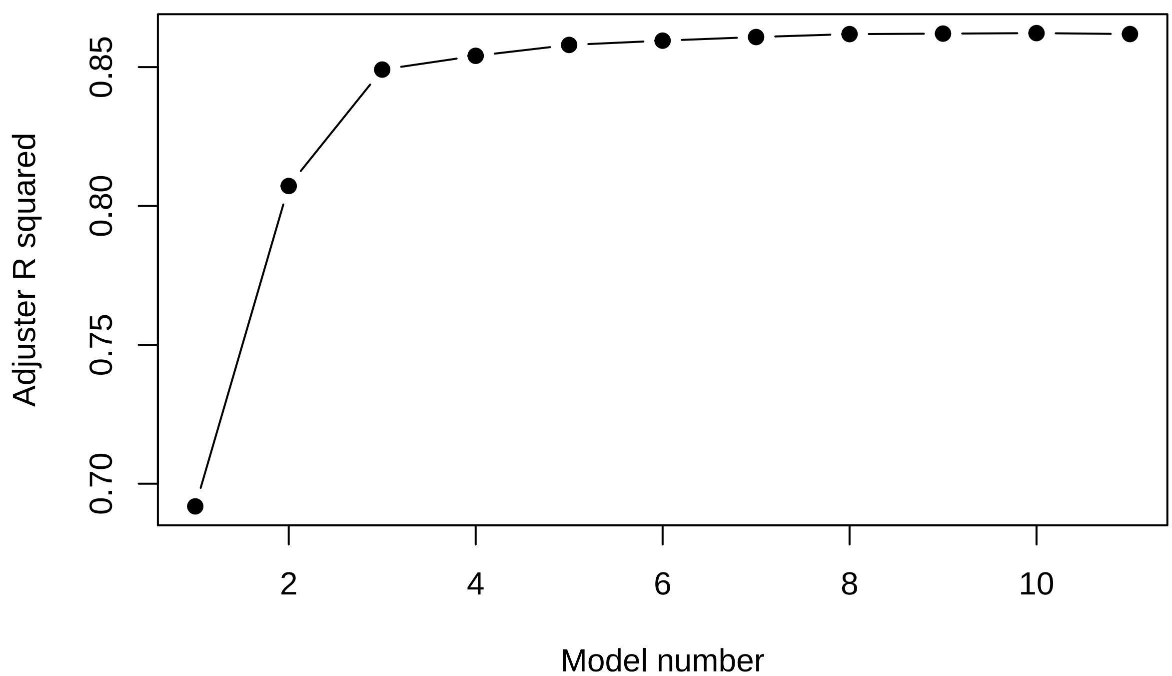
After we fit the 6 different models, we will compare model metrics and look at which model performed the best on this dataset.

Fitting Different Models

(a) Standard Least Squares We fit the data using the usual least-squares method. From a previous analysis, we know that we require quadratic terms for horsepower, displacement, and weight.

(b) Best-subset selection On using best-subset selection, we see that once the number of variables is more than 3, the increase in R^2 is not significant. We will then go ahead and fit the model with 3 vairables. This model has the `year` variable, and two terms of the `weight` variable.

```
#plot to see how many variables to pick
best.sub.adjr2 = summary(best.sub)$adjr2
plot(best.sub.adjr2, pch = 19, type = "b", xlab = "Model number", ylab = "Adjuster R squared", col = 1)
```

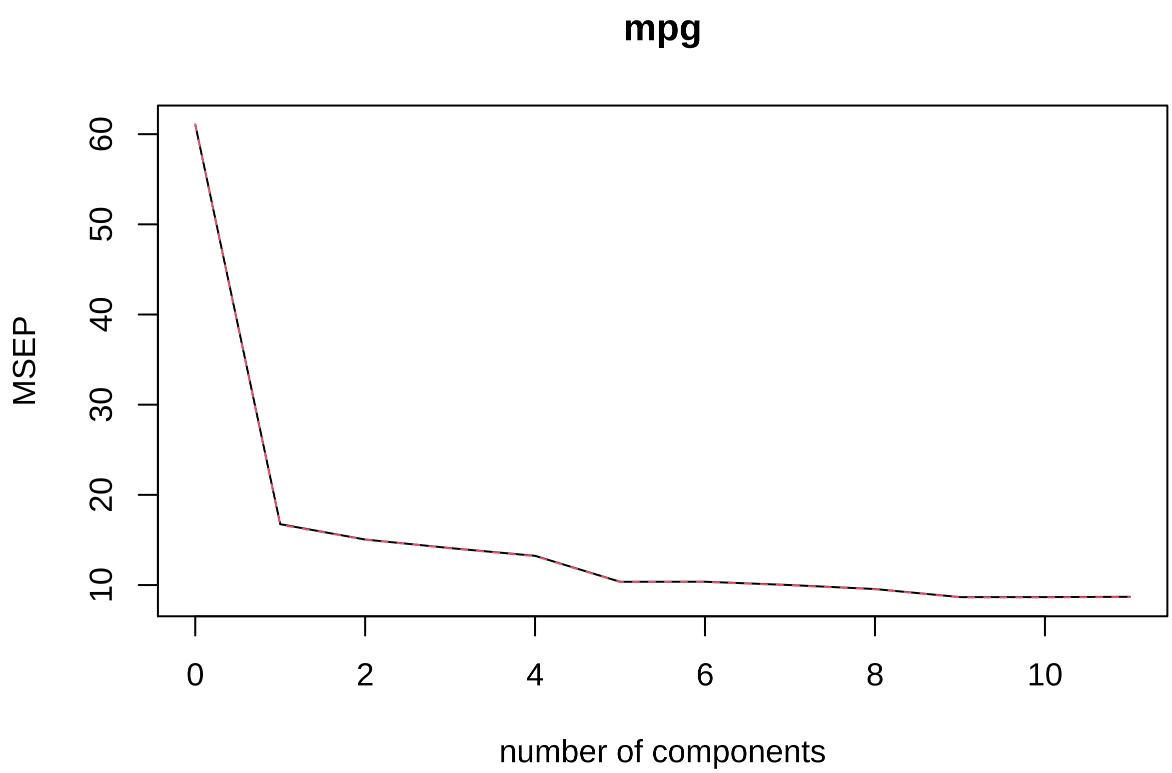


(c) Ridge regression On imposing the ridge penalty, we expect that the predictors will be shrunken significantly. We will compare them with the usual least square predictors at the end.

(d) Lasso regularization On imposing the lasso penalty, we know that variable selection kicks in. Once again, we will compare coefficients in the end to see which variables have been selected.

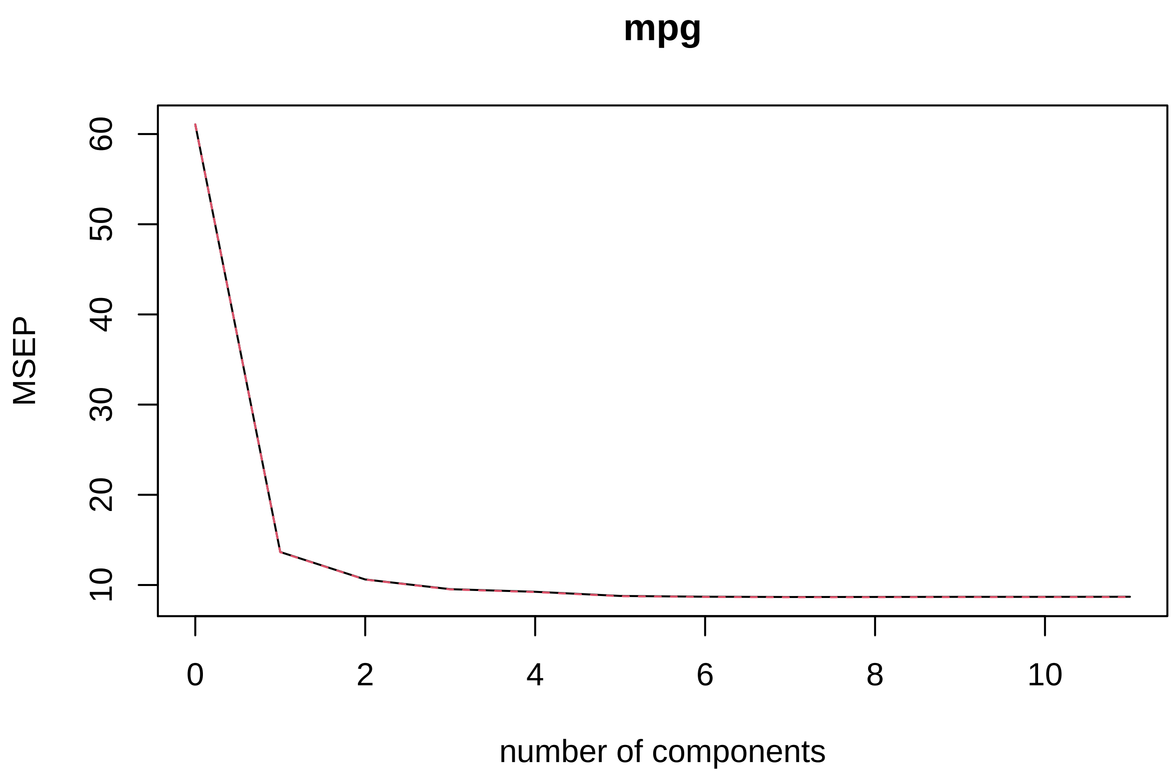
(e) Principal Component Regression (PCR) When we fit the model using PCR, we see that using the first 9 components yields the lowest error rate (computed from cross-validation).

```r
#Look at the MSEP plot
validationplot(pcr.fit, val.type = "MSEP")
```



(f) Partial Least Squares (PLS) When we fit the model using PLS, we see that using the first 7 components yields the lowest error rate (computed from cross-validation).

```r
#Look at the MSEP plot
validationplot(pls.fit, val.type = "MSEP")
```



Comparision

Finally, we take a look at the coefficients from the 6 methods we used in (a)-(f). We also list the test error rates - computed through cross-validation - to see which method performs the best. All the MSE's (Test Error) were calculated using 10-fold cross-validation.

Term	LS	Best Subset	Ridge	Lasso	PCR	PLS
Intercept	-35.814	-39.536	-28.111	-35.294		
cylinders	0.347		-0.079	0.192	0.693	0.734
poly(displacement, 2)1	-5.217		-17.974		-0.665	-0.661
poly(displacement, 2)2	9.672		10.185	8.535	0.606	0.453
poly(horsepower, 2)1	-43.683		-37.837	-40.883	-2.777	-2.459
poly(horsepower, 2)2	18.315		18.423	17.476	1.063	1.074
poly(weight, 2)1	-71.146	-109.779	-48.931	-73.207	-2.900	-3.157
poly(weight, 2)2	15.645	32.047	13.333	16.432	0.659	0.739
acceleration	-0.163		-0.132	-0.136	-0.700	-0.621
year	0.783	0.828	0.706	0.781	2.819	2.854
as.factor(origin)2	1.137		0.788	1.137	0.383	0.443
as.factor(origin)3	1.217		1.246	1.219	0.495	0.536
Test Error	8.720	9.300	8.896	8.808	8.741	8.677

The best-subset method gave us the simplest model, but this came at a cost. It also has the highest MSE. We see that the model from PLS gave us the lowest MSE at 8.677. I would recommend the model from (f) which was fitted using PLS since it has the lowest error rate among all 6 methods that we fitted.