CS4342-HW5

Ivan Martinovic

December 2021

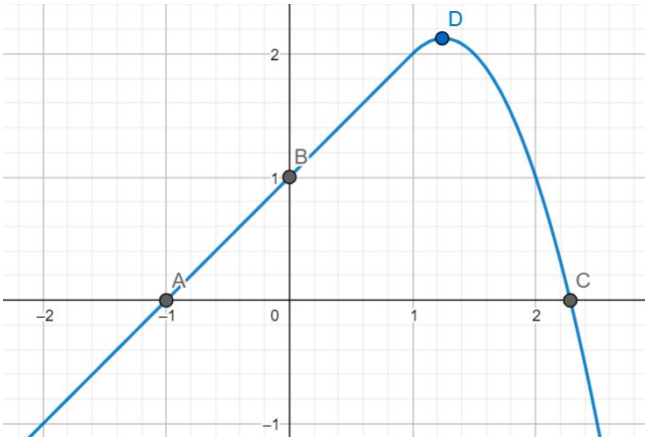
1 Applied Question

1.1 Question 1

Suppose we fit a curve with basis functions $b_1(X) = X$, $b_2(X) = (X-1)^2 I(X \geq 1)$. (Note that $I(X \geq 1)$ equals 1 for $X \geq 1$ and 0 otherwise.) We fit the linear regression model:

$$Y = \beta_0 + \beta_1 b_1(X) + \beta_2 b_2(X) + \epsilon$$

, and obtain coefficient estimates $\hat{\beta}_0 = 1, \hat{\beta}_1 = 1, \hat{\beta}_2 = -2$. Sketch the estimated curve between $X = -2$ and $X = 2$. Note the intercepts, slopes and other relevant information.



X-intercepts:

$$A = (-1, 0)$$

$$C = (\frac{5 + \sqrt{17}}{4}, 0) \approx (2.280, 0)$$

Y - intercept: $B = (0, 1)$

Maximum point: $D = (\frac{5}{4}, \frac{17}{8}) = (1.25, 2.125)$

Slopes are:

$$\frac{dY}{dX}_{X<1} = 1$$

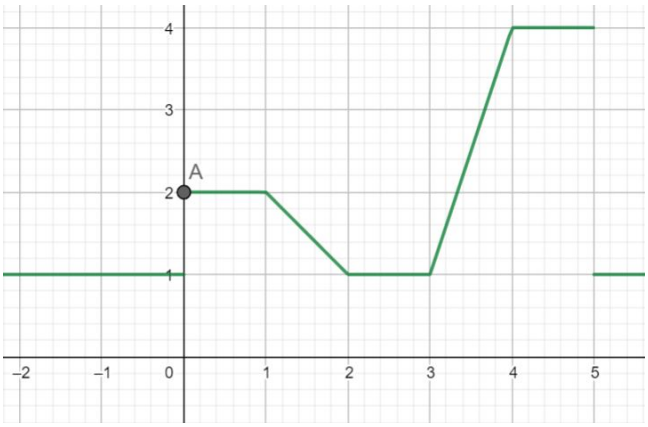
$$\frac{dY}{dX}_{X\geq 1} = 5 - 4X$$

1.2 Question 2

Suppose we fit a curve with basis functions $b_1(X) = I(0 \leq X \leq 2) - (X-1)I(1 \leq X \leq 2)$, $b_2(X) = (X-3)I(3 \leq X \leq 4) + I(4 < X \leq 5)$. We fit the linear regression model:

$$Y = \beta_0 + \beta_1 b_1(X) + \beta_2 b_2(X) + \epsilon$$

, and obtain coefficient estimates $\hat{\beta}_0 = 1, \hat{\beta}_1 = 1, \hat{\beta}_2 = 3$. Sketch the estimated curve between $X = -2$ and $X = 2$. Note the intercepts, slopes and other relevant information.



There are no X-intercepts.

Y - intercept: $A = (0, 2)$

Function is constant at intervals

$$X < 0, Y = 1$$

$$0 \leq X \leq 1, Y = 2$$

$$2 \leq X \leq 3, Y = 1$$

$$4 < X \leq 5, Y = 4$$

$$X > 5, Y = 1$$

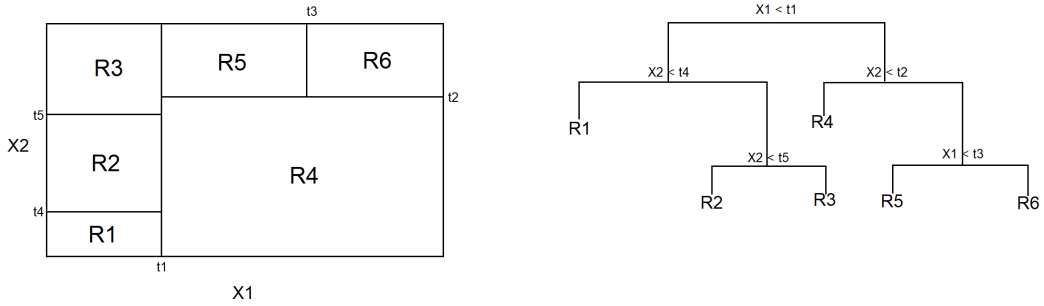
At remaining intervals the slopes are:

$$\frac{dY}{dX}_{1\leq X\leq 2} = -1$$

$$\frac{dY}{dX}_{3\leq X\leq 4} = 3$$

1.3 Question 3

Draw an example (of your own invention) of a partition of two dimensional feature space that could result from recursive binary splitting. Your example should contain at least six regions. Draw a decision tree corresponding to this partition. Be sure to label all aspects of your figures, including the regions R1, R2, . . ., the cutpoints t1, t2, . . ., and so forth.



1.4 Question 4

It is mentioned in Section 8.2.3 that boosting using depth-one trees (or stumps) leads to an additive model: that is, a model of the form:

$$f(X) = \sum_{j=1}^{p} f_j(X_j)$$

Explain why this is the case. You can begin with (8.12) in Algorithm 8.2.

Equation (8.12) is the following:

$$\hat{f}(X) = \sum_{b=1}^{B} \lambda f^b(X)$$

f^b is a tree with d splits (d+1 terminal nodes) which is fit to X and residuals of the tree f^{b-1}.

By using only one split (i.e. $d = 1$), we are guaranteeing that $f^b(X)$ is of the form:

$$f^b(X) = I(X_j < t_b) * C_1 + I(X_j \geq t_b) * C_2 = f_{ji}(X_j)$$

,where X_j is the j-th predictor, t_b is the cutoff point for the b-th tree, C_1 and C_2 are constants and $f_{ji}(X_j)$ is the i-th function involving the j-th predictor. Substituting this into (8.12):

$$\hat{f}(X) = \sum_{j=1}^{p} \sum_{i=1}^{I_j} f_{ji}(X_j)$$

Where p is the total number of predictors, and I_j is the total number of stumps whose function involves the j-th predictor (and since $d = 1$ only the j-th predictor).
We can now aggregate the inner sum as just a single function of X_j, $f_j(X_j)$:

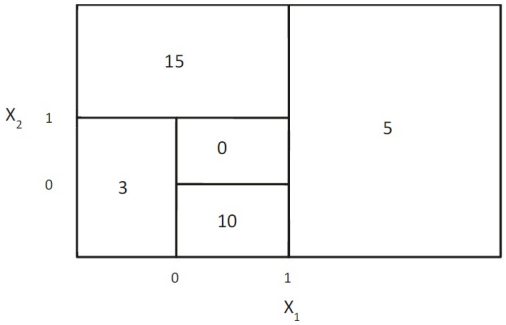
$$\hat{f}(X) = \sum_{j=1}^{p} f_j(X_j)$$

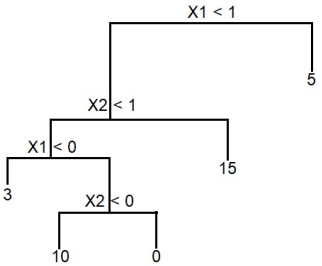
q.e.d.

1.5 Question 5

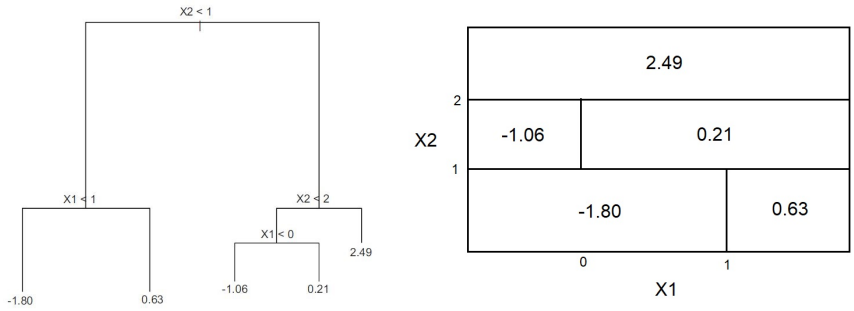
This question relates to the plots in the below figures.

(a) Sketch the tree corresponding to the partition of the predictor space illustrated in the left-hand panel of the below figure. The numbers inside the boxes indicate the mean of Y within each region.





(b) Create a diagram similar to the left-hand panel of the figure, using the tree illustrated in the right-hand panel of the same figure. You should divide up the predictor space into the correct regions, and indicate the mean for each region.



1.6 Question 6

Suppose we produce ten bootstrapped samples from a data set containing red and green classes. We then apply a classification tree to each bootstrapped sample and, for a specific value of X, produce 10 estimates of P(Class is Red—X):

0.1, 0.15, 0.2, 0.2, 0.55, 0.6, 0.6, 0.65, 0.7, and 0.75.

There are two common ways to combine these results together into a single class prediction. One is the majority vote approach discussed in this chapter. The second approach is to classify based on the average probability. In this example, what is the final classification under each of these two approaches?

Using Majority vote we have that 6 out of 10 estimates have a probability over 0.5, which implies they predict the response is Red. Since 6 out of 10 is a majority, the final class prediction is Red.

Using average of probabilities we have:

$$P(Class = Red|X) = \frac{0.1 + 0.15 + 0.2 + 0.2 + 0.55 + 0.6 + 0.6 + 0.65 + 0.7 + 0.75}{10} = \frac{4.5}{10} = 0.45$$

Since there are only two classes (i.e. Red and Green), this implies that

$$P(Class = Green|X) = 0.55$$

, i.e. the final class prediction is Green.

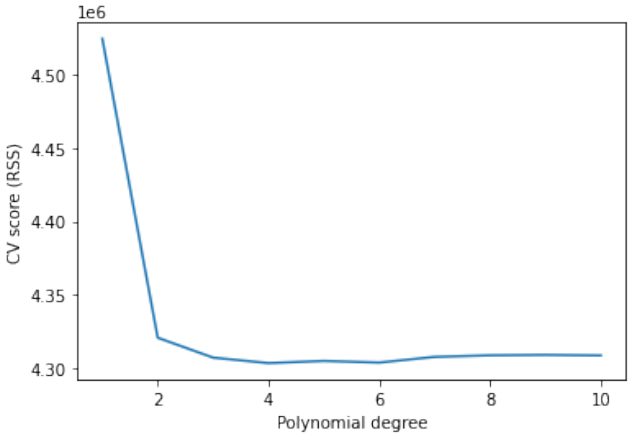
2 Applied Questions

2.1 Question 1

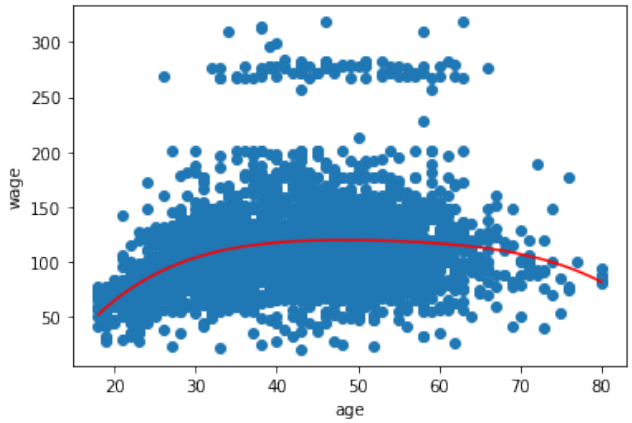
In this exercise, we will further analyze the Wage data set.

(a) Perform polynomial regression to predict wage using age. Use cross-validation to select the optimal degree d for the polynomial. Make a

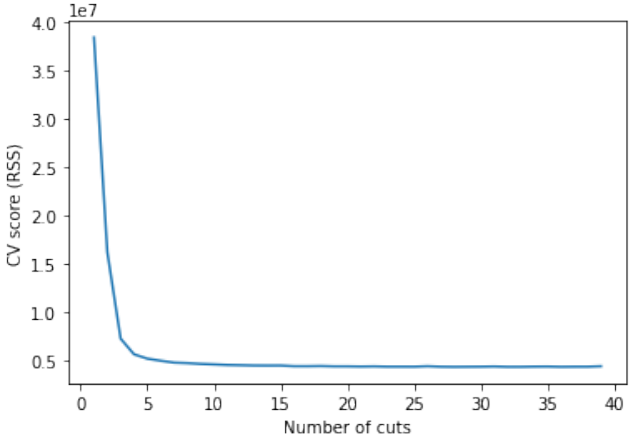
plot of the resulting polynomial fit to the data.
The following is a graph of the cross-validation scores vs the polynomial degrees:



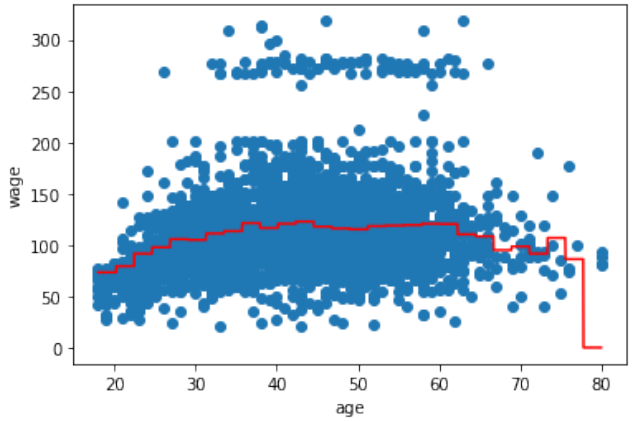
The best-degree polynomial is the quartic (degree 4) polynomial. The fit using the quartic polynomial is shown below:



(b) Fit a step function to predict wage using age, and perform cross-validation to choose the optimal number of cuts. Make a plot of the fit obtained.
The following is a graph of the cross-validation scores vs the number of cuts:



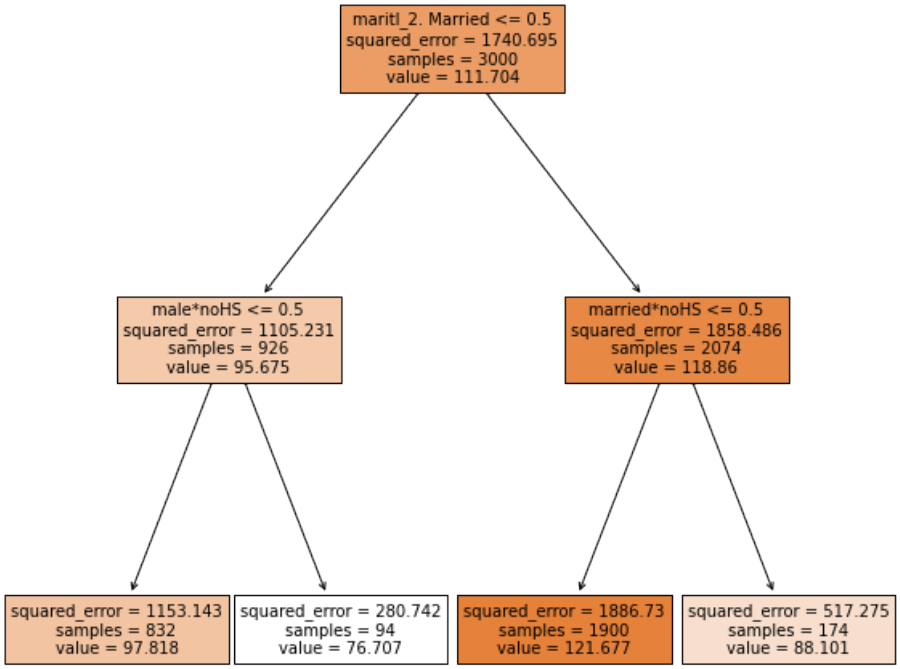
The best step function contains 28 cuts. The fit using 28 cuts is shown below:



2.2 Question 2

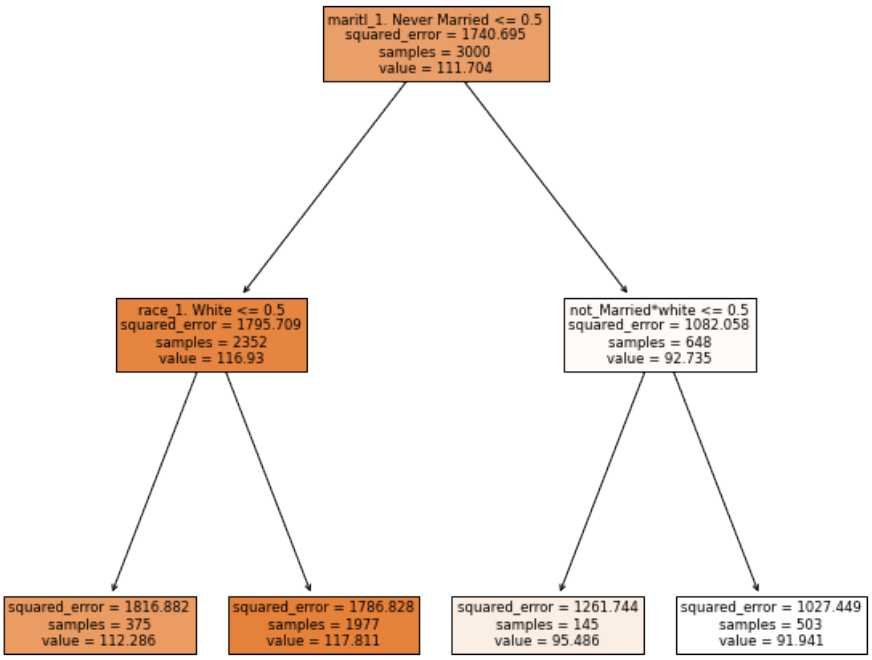
The Wage data set contains a number of other features not explored in Chapter 7, such as marital status (maritl), job class (jobclass), and others. Explore the relationships between some of these other predictors and wage, and use non-linear fitting techniques in order to fit flexible models to the data. Create plots of the results obtained, and write a summary of your findings.

Here is a regression tree using predictors whether a person is married, male and has no highschool education, as well as their interactions:



As we can see, the first most important predictor is whether the person is married. On average married people have a higher wage than non-married people. If someone is not married, the most important predictor is the combined predictor whether a person is both a male and has no high school education. We see that people who are both male and have no high school education have a lower wage than people who do not posses those predictors. If the person is married the most important predictor becomes whether someone has no high school education. We see again that people with no high school education have on average a lower salary. By looking at the mean squared errors, we see that we can make the most accurate prediction about someones wage if they are not married, male and with no high school education.

Here is another regression tree using predictors whether the person has never married, white, male and their interactions:

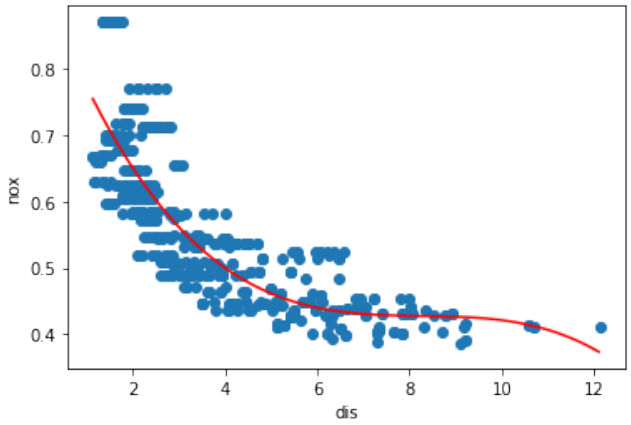


Here the most important predictor is whether someone never married. We see that people who never married have a lower salary than those who did. The next most important predictor becomes whether a person is white. On average it seems that, if a person is white they have a lower salary. We can make the most accurate prediction when someone has never married and is white because the squared error is the lowest.

2.3 Question 3

This question uses the variables dis (the weighted mean of distances to five Boston employment centers) and nox (nitrogen oxides concentration in parts per 10 million) from the Boston data. We will treat dis as the predictor and nox as the response.

(a) Fit a cubic polynomial regression to predict nox using dis. Report the regression output, and plot the resulting data and polynomial fits.
The following fit was obtained:



The coefficients are as follows:

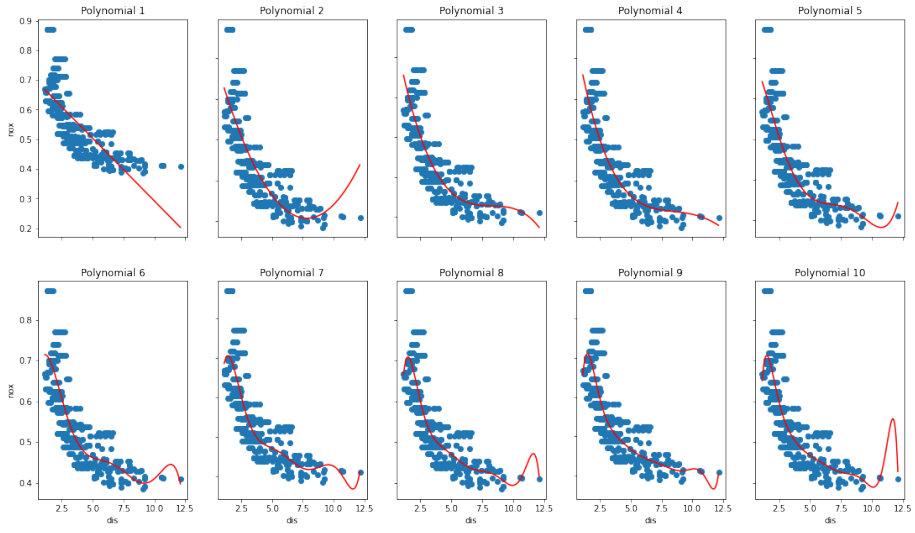
$$\beta_0 = 0.9341280720211882$$

$$\beta_1 = -0.18208169$$

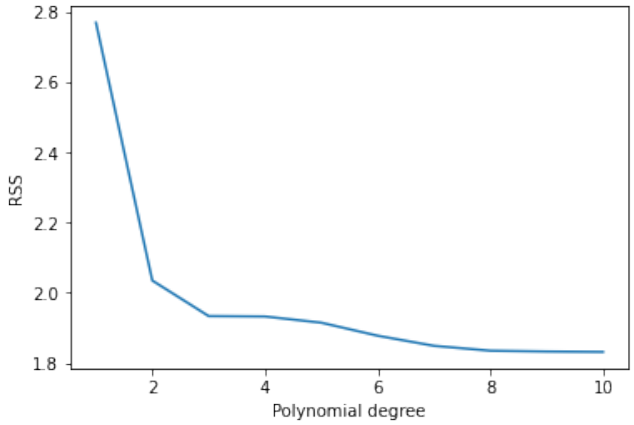
$$\beta_2 = 0.02192766$$

$$\beta_3 = -0.000885$$

(b) Plot the polynomial fits for a range of different polynomial degrees (say, from 1 to 10), and report the associated residual sum of squares.
The following polynomial fits were obtained:



The graphs of each polynomial degree and their RSS is shown below:



The residual sum of squares are:

$$RSS_1 = 2.768562858969276$$

$$RSS_2 = 2.0352618689352564$$

$$RSS_3 = 1.9341067071790705$$

$$RSS_4 = 1.9329813272985938$$

$$RSS_5 = 1.9152899610843037$$

$$RSS_6 = 1.878257298508164$$

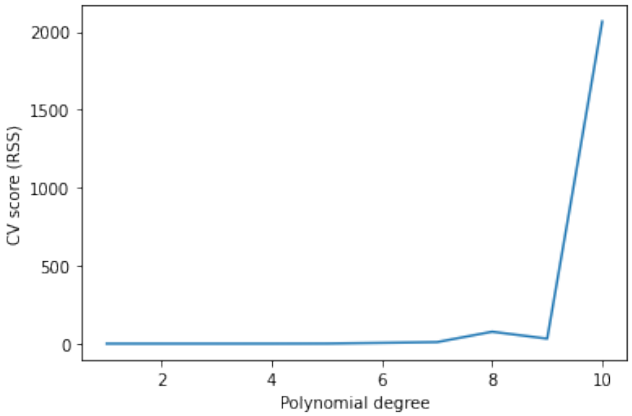
$$RSS_7 = 1.849483614582971$$

$$RSS_8 = 1.8356296890678305$$

$$RSS_9 = 1.8333308045166714$$

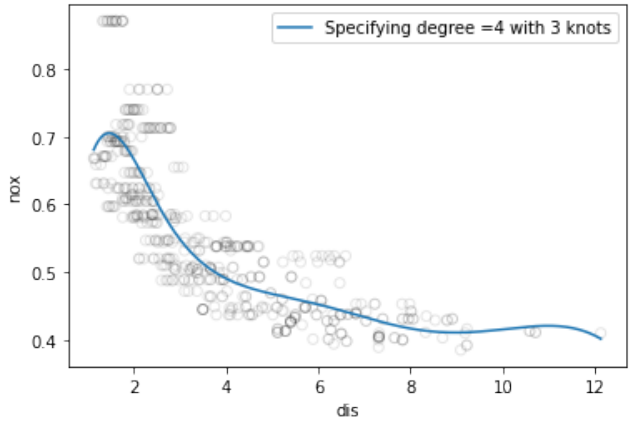
$$RSS_{10} = 1.832171125206099$$

(c) Perform cross-validation or another approach to select the optimal degree for the polynomial, and explain your results.
Using cross-validation the following CV-scores vs polynomial degree plot was obtained:



Although it is hard to distinguish, the numbers say that polynomial of 3rd degree is best. Polynomials of degree 1 through 5 have a CV-score in the range between 0.25 and 0.7, implying that they would likely be a good fit for the true relationship, as they optimize the bias-variance trade-off. Degrees above 6, are too flexible; they overfit the data making their variance high and also their CV score.

(d) Fit a regression spline to predict nox using dis. Report the output for the fit using four degrees of freedom. How did you choose the knots? Plot the resulting fit.
Here is the spline with 4 degrees of freedom with cutpoints at 3 and 7:



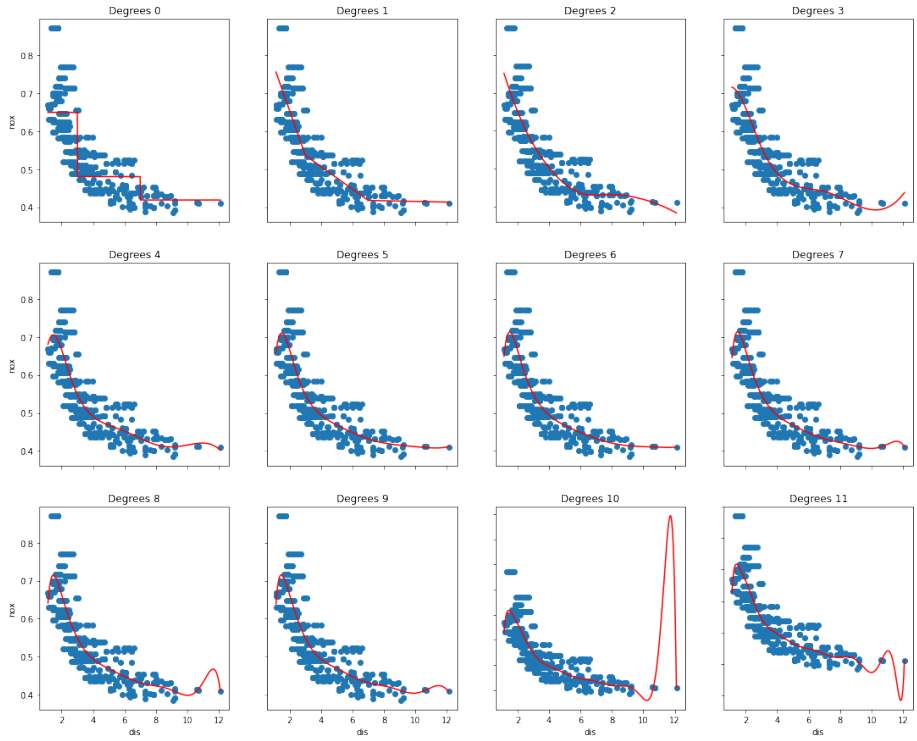
The knots were chosen by visually inspecting the data and placing knots in places where it was felt the function might vary most rapidly.
Here are the summary tables for the fit:

Generalized Linear Model Regression Results

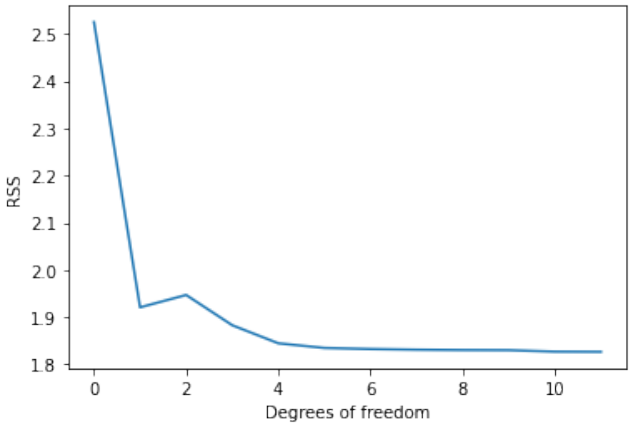
Dep. Variable:	nox	**No. Observations:**	506
Model:	GLM	**Df Residuals:**	499
Model Family:	Gaussian	**Df Model:**	6
Link Function:	identity	**Scale:**	0.0036966
Method:	IRLS	**Log-Likelihood:**	702.43
Date:	Mon, 06 Dec 2021	**Deviance:**	1.8446
Time:	02:42:09	**Pearson chi2:**	1.84
No. Iterations:	3		
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	0.6808	0.019	35.551	0.000	0.643	0.718
bs(X['dis'], knots=(3, 7), degree=4, include_intercept=False)[0]	0.0753	0.030	2.495	0.013	0.016	0.134
bs(X['dis'], knots=(3, 7), degree=4, include_intercept=False)[1]	-0.2441	0.023	-10.476	0.000	-0.290	-0.198
bs(X['dis'], knots=(3, 7), degree=4, include_intercept=False)[2]	-0.1527	0.056	-2.740	0.006	-0.262	-0.043
bs(X['dis'], knots=(3, 7), degree=4, include_intercept=False)[3]	-0.3459	0.057	-6.036	0.000	-0.458	-0.234
bs(X['dis'], knots=(3, 7), degree=4, include_intercept=False)[4]	-0.2277	0.070	-3.236	0.001	-0.366	-0.090
bs(X['dis'], knots=(3, 7), degree=4, include_intercept=False)[5]	-0.2792	0.062	-4.540	0.000	-0.400	-0.159

(e) Now fit a regression spline for a range of degrees of freedom, and plot the resulting fits and report the resulting RSS. Describe the results obtained. Here are the resulting fits ranging from 0 to 11 degrees of freedom:



The RSS vs degrees of freedom is show below:



The resulting residual sums of squares are:

$$RSS_0 = 2.5247281883591177$$

$$RSS_1 = 1.92072506480702$$

$$RSS_2 = 1.9469257679514318$$

$$RSS_3 = 1.882849524286999$$

$$RSS_4 = 1.8445899533238388$$

$$RSS_5 = 1.8347679418373368$$

$$RSS_6 = 1.8324727064870916$$

$$RSS_7 = 1.831070711727015$$

$$RSS_8 = 1.8303023223656425$$

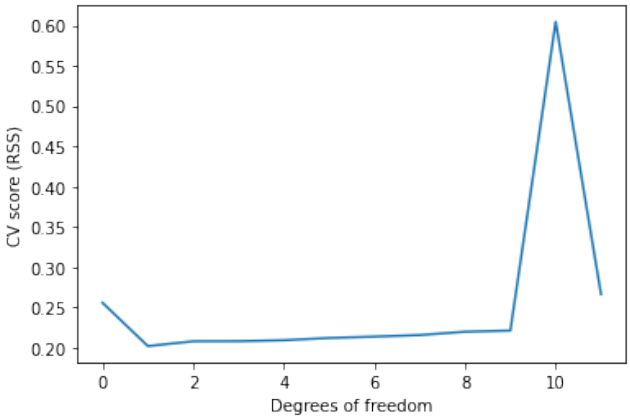
$$RSS_9 = 1.830004473764256$$

$$RSS_{10} = 1.8269670361043449$$

$$RSS_{11} = 1.8266123416787878$$

RSS decreases as the degrees of freedom increases. This is due to the fact that we are using the whole data set as the training set, and more flexible models (in this case ones with more degrees of freedom) will have a smaller RSS.

(f) Perform cross-validation or another approach in order to select the best degrees of freedom for a regression spline on this data. Describe your results.
Using cross-validation the following CV-scores vs degrees of freedom plot was obtained:



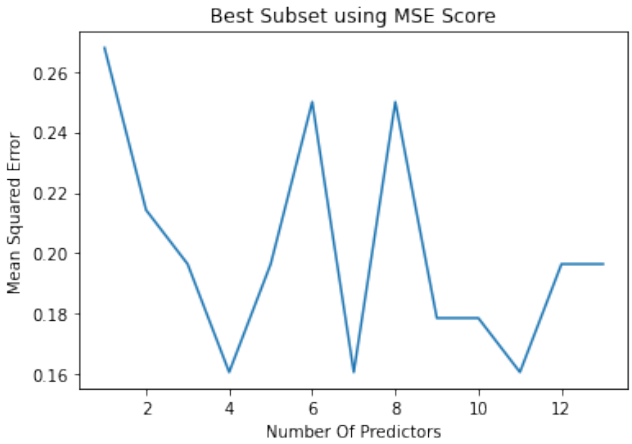
From the graph we see that the model with a single degree of freedom is the best as it has the lowest CV score (which is the RSS). Although we can see that models involving 1 to 9 degrees of freedom have similar CV scores ranging from 0.21 to 0.24, implying that any of these models would likely be accurate at representing the actual relationship between dis and nox. Using 0 degrees of freedom is too inflexible and has too high bias. Using more than 9 degrees of freedom leads to overfitting.

2.4 Question 4

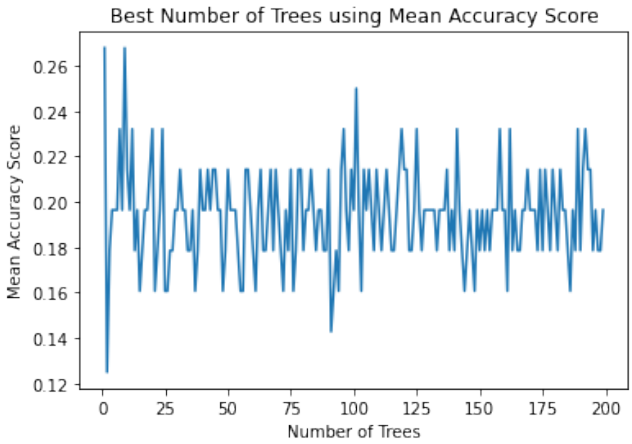
Apply random forests to predict mdev of the Boston data after converting it into a qualitative response variable – values above the median of mdev is set 1 and others are set to zero. Use all other predictors in preditction of the qualitative data using 25 and 500 trees. Create a plot displaying the test error resulting from random forests on this data set for a more comprehensive range of values of number of predictors and trees. Describe the results obtained.

Using 25 trees the test MSE 0.19642857142857142 Using 500 trees the test MSE was 0.17857142857142858

Using 30 trees, validation set approach and best subset selection the following number of predictors vs test MSE plot was obtained:



We see that the best subset has 4 predictors, and they are 'crim', 'rm', 'black' and 'lstat'
Note: These results are highly variable and dependent on the bootstrapping process, but it seems as though the test MSE decreases as we increase the number of predictors. This procedure also took a lot of time
Using validation set approach, and all predictors, the following tree number vs number test MSE plot was obtained:



The best number of trees used is apparently 2, althouth the variance is quite high, and results are highly dependent on the random state of the bootstrap methods. The variance also seems to decrease as the number of trees increases.

2.5 Question 5

We want to predict Sales in the Carseats data set using regression trees and related approaches.

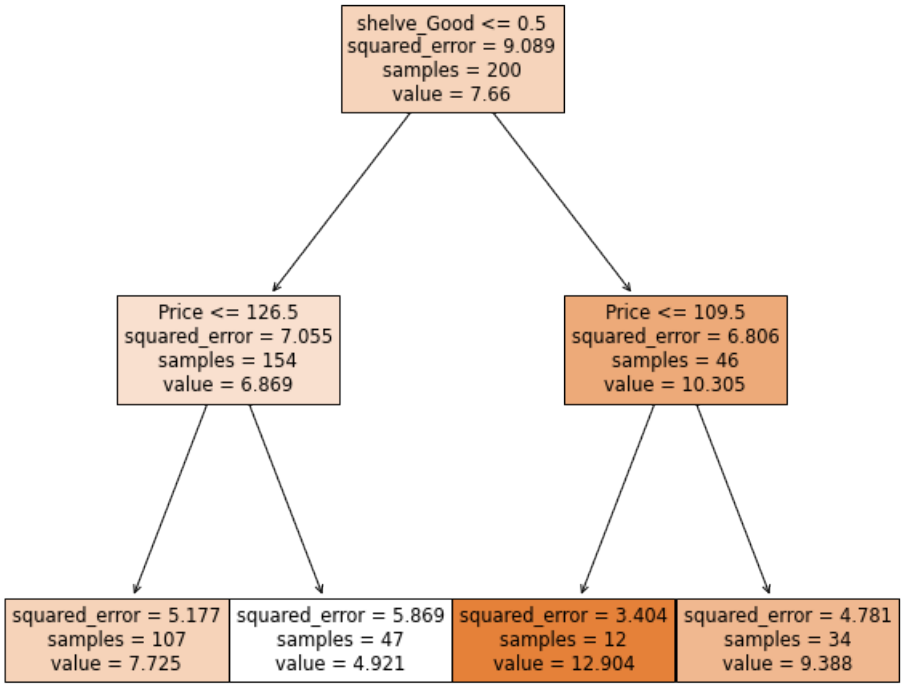
(a) Split the data set into a training set and a test set.

```
] from sklearn.model_selection import train_test_split
  from sklearn.tree import DecisionTreeRegressor
  from sklearn.metrics import mean_squared_error
  clf = DecisionTreeRegressor(max_depth=3)

  X_train, X_test, y_train, y_test = train_test_split(
       X, y, test_size=0.5, random_state=1234)
```

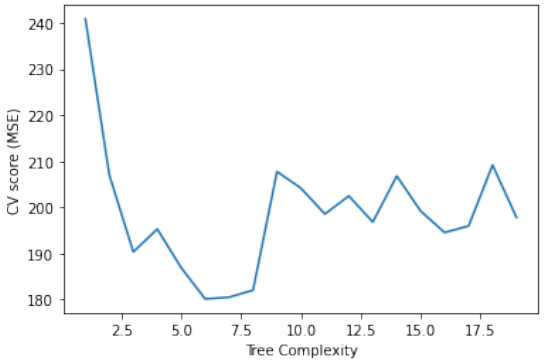
(b) Fit a regression tree to the training set. Plot the tree, and interpret the results. What test MSE do you obtain?

Below is diagram of a regression tree with depth 2 (deeper trees could not be displayed in an aesthetically pleasing fashion as they were too large and too complex).
From the tree we see that the first most important predictor is the "shelve" dummy variable, followed by the "price" variable.
If the dummy variable is 0 and price is less than 126.5, then sales are 7.725.
If the dummy variable is 0 and price is greater than 126.5, then sales are 4.921.
If the dummy variable is 1 and price is less than 109.5, then sales are 12.904.
If the dummy variable is 1 and price is greater than 109.5, then sales are 9.388.

The MSE is: 4.803816271935667



(c) Use cross-validation in order to determine the optimal level of tree complexity. Does pruning the tree improve the test MSE? Using cross-validation the following MSE vs. tree complexity plot was obtained:



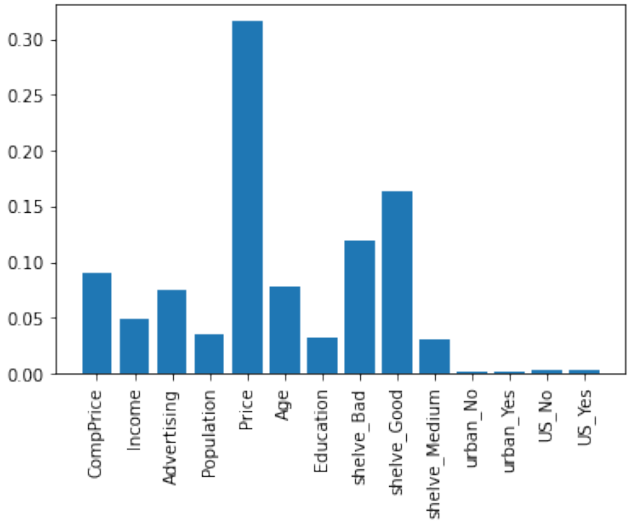
As we can see, pruning did improve test MSE because optimal level of tree complexity is 6.

(d) Use the bagging approach in order to analyze this data. What test MSE do you obtain? Determine which variables are most important (variable impooratnce measure).

Using bagging the test MSE obtained was: 2.4321834813999987
Note: The bagging regressor does not have feature_importances_ attribute, so the variable importance plots could not be drawn

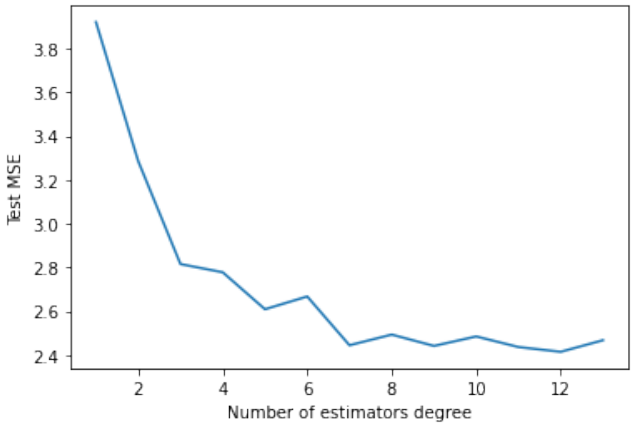
(e) Use random forests to analyze this data. What test MSE do you obtain? Determine which variables are most important (variable importance measure). Describe the effect of m, the number of variables considered at each split, on the error rate obtained.]

Using random forests the test MSE obtained was 2.385195091199999. The bar plot of most important variables is shown below:



As we may see the most important predictors are "Price", "shelve_Good" and "sheve_Bad" dummy variables, "CompPrice" and etc.

Below is a plot of m against the test MSE:



As we can see the test MSE decreases as we consider more predictors at each split, reaching a minimum when m=12.

2.6 Question 6

We now use boosting to predict Salary in the Hitters data set.

(a) Remove the observations for whom the salary information is unknown, and then log-transform the salaries.

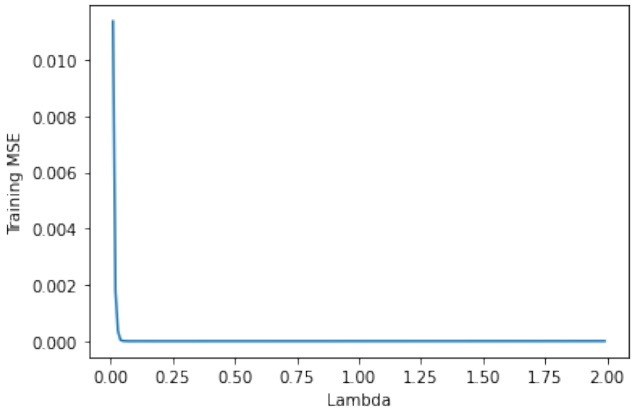
```
import numpy as np
import pandas as pd

data = pd.read_csv("Hitters.csv", na_values="NA").dropna()


leagueDummies = pd.get_dummies(data['League'], prefix="League")
divisionDummies = pd.get_dummies(data['Division'], prefix='Division')
NewLeagueDummies = pd.get_dummies(data['NewLeague'], prefix='NewLeague')

data = data.join(leagueDummies).join(divisionDummies).join(NewLeagueDummies).drop(['League'], axis=1).drop(['Division'], axis=1).drop(['NewLeague'],

data['Salary'] = np.log(data['Salary'])
```

(b) Create a training set consisting of the first 200 observations, and a test set consisting of the remaining observations.

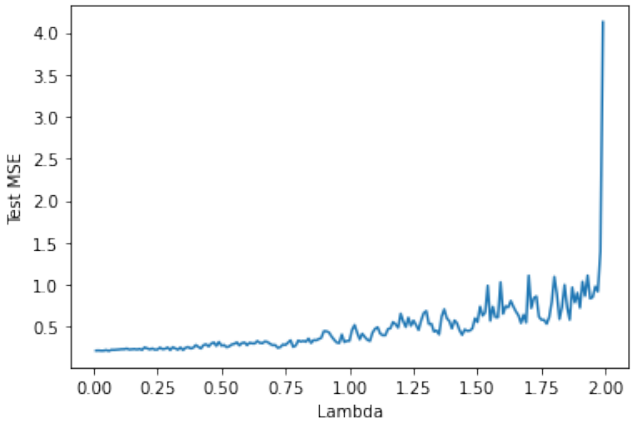
```
2] dataTrain = data.iloc[0:200]
   dataTest = data.iloc[200:]

   X_train = dataTrain.drop(['Salary'], axis=1)
   X_test = dataTest.drop(['Salary'], axis=1)

   y_train = dataTrain['Salary']
   y_test = dataTest['Salary']
```

(c) Perform boosting on the training set with 1,000 trees for a range of values of the shrinkage parameter λ. Produce a plot with different shrinkage

values on the x-axis and the corresponding training set MSE on the y-axis.



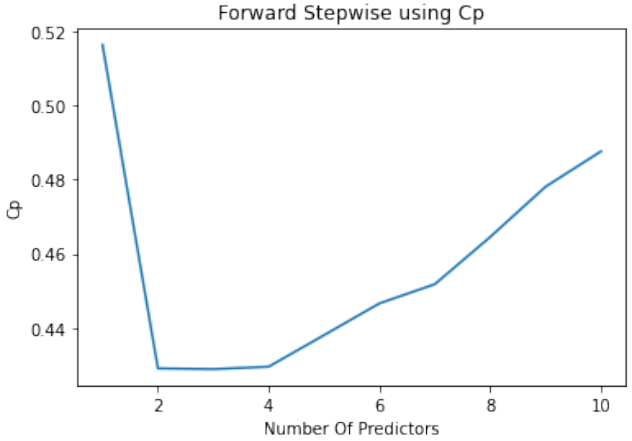
(d) Produce a plot with different shrinkage values on the x-axis and the corresponding test set MSE on the y-axis.



(e) Compare the test MSE of boosting to the test MSE that results from applying two of the regression approaches seen in Chapters 3 and 6.

Test MSE from boosting (using $lambda = 0.01$) =0.21818894725763538
Test MSE using simple linear regression with all predictors = 0.49179593754548967

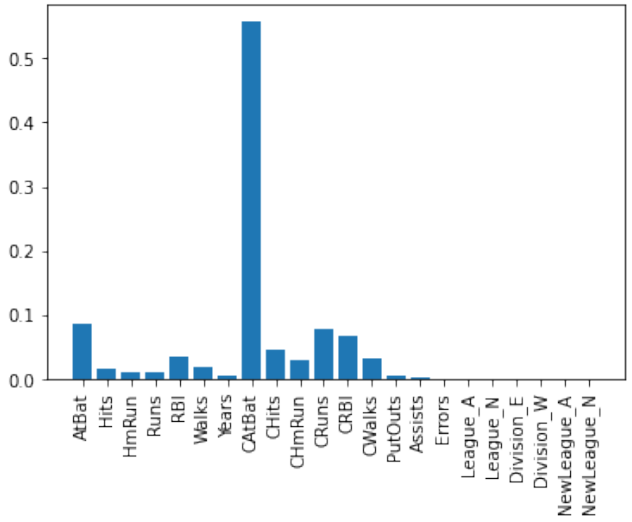
Here is a graph of test MSE and number of predictors for the forward stepwise selection.



Test MSE using simple linear regression with best number of predictors (using forward stepwise selection) = 0.47442731844353336

As we can see, boosting has the smallest test MSE.

(f) Which variables appear to be the most important predictors in the boosted model?
Below is a bar plot of the importance of variables in the boosted model:



As we can see, the most important predictors are CATBat, AtBat, CRuns, CRBI, etc.

(g) Now apply bagging to the training set. What is the test set MSE for this approach?

The test MSE using Bagging is 0.24156594881485402. It is lower than form methods from Chapters 3 and 6, but still higher than for the boosted model.