CS4342-HW4

Ivan Martinovic

November 2021

1 Conceptual and theoretical questions

1. Using basic statistical properties of the variance, as well as single-variable calculus, derive (1). In other words, prove that α given by (1) does indeed minimize $Var(\alpha X + (1-\alpha)Y)$

$$\alpha = \frac{{\sigma_Y}^2 - \sigma_{XY}}{{\sigma_X}^2 + {\sigma_X}^2 - 2\sigma_{XY}} \qquad ...(1)$$

Let's start with the first equation:

$$Var(\alpha X + (1-\alpha)Y)$$

We use the property of variance: $Var(X) = E(X^2) - (E(X))^2$, and substitution $Z = \alpha X + (1-\alpha)Y$:

$$Var(\alpha X + (1-\alpha)Y) = Var(Z) = E(Z^2) - (E(Z))^2 \text{ ... } (2)$$

Let's first evaluate $E(Z^2)$:

$$E(Z^2) = E(\alpha^2 X^2 + (1-\alpha)^2 Y^2 + 2\alpha(1-\alpha)XY)$$

Next we apply two rules of expected values: 1) $E(aX) = a * E(X)$ and 2) $E(X+Y) = E(X) + E(Y)$:

$$E(Z^2) = \alpha^2 E(X^2) + (1-\alpha)^2 E(Y^2) + 2\alpha(1-\alpha)E(XY) \text{ ... } (3)$$

We now evaluate $(E(Z))^2$, using the same two rules for expected values:

$$(E(Z))^2 = (E(\alpha X + (1-\alpha)Y))^2 = (\alpha E(X) + (1-\alpha)E(Y))^2$$
$$(E(Z))^2 = \alpha^2 E(X)^2 + (1-\alpha)^2 E(Y)^2 + 2\alpha(1-\alpha)E(X)E(Y) \qquad ...(4)$$

We now combine results from (3) and (4) and substitute them into (2):

$$Var(Z) = \alpha^2(E(X^2) - E(X)^2) + (1-\alpha)^2(E(Y^2) - E(Y)^2) + 2\alpha(1-\alpha)(E(XY) - E(X)E(Y))$$

We finally use expressions $Var(X) = E(X^2) - (E(X))^2$ and $Cov(XY) = E(XY) - E(X)E(Y)$ to obtain:

$$Var(Z) = \alpha^2 {\sigma_X}^2 + (1-\alpha)^2 {\sigma_Y}^2 + 2\alpha(1-\alpha)\sigma_{XY} = f...(5)$$

We now take the derivative of (5) with respect to α and set it to 0, to find the α which minimizes $Var(\alpha X + (1-\alpha)Y)$:

$$\frac{df}{d\alpha} = 2\alpha {\sigma_X}^2 + 2(\alpha - 1){\sigma_Y}^2 + (2 - 4\alpha)\sigma_{XY} = 0$$

Divide both sides by 2, and get α on the left hand side:

$$\alpha({\sigma_X}^2 + {\sigma_Y}^2 - 2\sigma_{XY}) = {\sigma_Y}^2 - \sigma_{XY}$$

$$\alpha = \frac{{\sigma_Y}^2 - \sigma_{XY}}{{\sigma_X}^2 + {\sigma_Y}^2 - 2\sigma_{XY}} = (1)$$

q.e.d.

2. We now review k-fold cross-validation.

   (a) Explain how k-fold cross-validation is implemented.
   In k-fold cross-validation the data set is divided into a k number of equal data sets, called "folds" which are of roughly equal size. The first set is treated as the test or validation set, while the remaining k-1 sets are treated as training sets to which the model is fitted. Once the model is fitted, we use the validation set to calculate the test mean squared error or MSE. This procedure of choosing one set as the validation set, and all others as training sets; fitting the model and calculating the MSE is repeated for the remaining k-1 folds. Finally a mean of all the MSE's is calculated which is our cross validation estimate of the true MSE of the model.

   (b) What are the advantages and disadvantages of k-fold cross validation relative to:

      (i) The validation set approach?
      Advantages: Estimates of the test error are less variable than the validation set approach. The model is fit on k-1 / k proportion of the data set using k-fold cross-validation, rather than half the data set using the validation set approach, which makes it less biased relative to the training data set, and also it does not overestimate the test error rate as much.
      Disadvantages: Slightly harder to implement and conceptually is slightly more complex.

(ii) LOOCV? Advantages: LOOCV suffers from performance issues since the model has to be fit n times, whereas in K-fold CV, the model only needs to be fitted k times (where k is typically 5 or 10). The test error rate of k-fold CV also has smaller variance than that of LOOCV, since the output values of test errors of k-fold CV are less correlated than LOOCV. By the bias-variance trade-off k-fold CV can therefore sometimes give better estimates for the test error rate in some cases. Disadvantages: The test error rate of LOOCV has less bias, since the training data sets include n-1 observations, whereas the k-fold CV training data sets include n-k/n observation. Again, depending on the bias-variance trade-off this can lead LOOCV to have better estimates of the test error rate in some cases.

3. Suppose that we use some statistical learning method to make a prediction for the response Y for a particular value of the predictor X. Carefully describe how we might estimate the standard deviation of our prediction.

Suppose our training data set contains N observations. We can use the Bootstrap method to draw n different bootstrap samples (each containing N observations) from our training data set, and fitting our model to those samples, which will produce n models. We use all of these model fits to make n predictions for Y for the particular predictor X. We then compute the standard deviation of these n predictions.
We could have also done an estimate for the standard deviation using K-fold cross validation. Every single time we fit a model to the (k-1) folds, we could also use this model fit to make a prediction for Y using the particular predictor X. By the end, we would have k estimates for Y, from which we can compute the standard deviation.

4. We perform best subset, forward stepwise, and backward stepwise selection on a single data set. For each approach, we obtain p + 1 models, containing 0, 1, 2, . . . , p predictors. Explain your answers:

(a) Which of the three models with k predictors has the smallest training RSS?
The best subset selection will have the smallest training RSS. Based on how it is defined, at step k, the best subset selection looks at all models containing k predictors, and selects the one model whose k predictors give the lowest training RSS. Therefore the best subset selection is guaranteed to have the lowest training RSS for a model containing k predictors; whereas forward and backward stepwise selection methods, beacause they are greedy algorithms, may or may-not contain the k predictors in their model which produce the lowest training RSS.

(b) Which of the three models with k predictors has the smallest test RSS?
The answer to this question depends on the total number of predictors p in our data set.

If the number of predictors is small, we expect best subset selection to have the lowest test RSS since it is always guaranteed to produce as good or better model than forward or backward stepwise selection.
However, if the number of predictors is very large, then best subset selection may suffer from high variance. The larger the search space, the higher the chance of finding models that look good on the training data, even though they might not have any predictive power on future data. Thus an enormous search space can lead to overfitting and high variance of the coefficient estimates.

(c) True or False:

i. The predictors in the k-variable model identified by forward stepwise are a subset of the predictors in the (k+1)-variable model identified by forward stepwise selection.
True - The (k+1)-variable model using forward stepwise selection is constructed by adding 1 predictor to the k)-model using forward stepwise selection, which results in a model with lowest RSS.

ii. The predictors in the k-variable model identified by backward stepwise are a subset of the predictors in the (k + 1)- variable model identified by backward stepwise selection.
True - The k-variable model using backward stepwise selection is constructed by removing 1 predictor from the (k+1)-model using backward stepwise model, which results in a model with lowest RSS.

iii. The predictors in the k-variable model identified by backward stepwise are a subset of the predictors in the (k + 1)- variable model identified by forward stepwise selection.
False - Example: if the best 1 and 2-predictor model include X_1, and X_1 and X_2 respectively and the best (n-1) and (n-2) predictor models do not include X_1, and X_1 and X_2 respectively, then the 1-variable model using backward stepwise selection cannot be a subset of the 2-variable model using forward stepwise selection.

iv. The predictors in the k-variable model identified by forward stepwise are a subset of the predictors in the (k+1)-variable model identified by backward stepwise selection.
False - Example: if the best 1-predictor model includes X_1, and the best (n-1) predictor model does not, then the (n-2)-variable model using forward stepwise selection cannot be a subset of the (n-1)-variable model using backward stepwise selection.

v. The predictors in the k-variable model identified by best subset are a subset of the predictors in the (k + 1)-variable model identified by best subset selection.
False - Example: Suppose the best 1-variable model contains X_1, but the best 2 variable model contains X_2 and X_3. Therefore the predictors in the 1-variable model are not a subsets of the predictors in the 2-variable model.

4. Suppose we estimate the regression coefficients in a linear regression model by minimizing:

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij}\right)^2 \text{ subject to } \sum_{j=1}^{p} |\beta_j| \leq s$$

for a particular value of s - s is positive. For parts (a) through (e), indicate which of i. through v. is correct. Justify your answer.

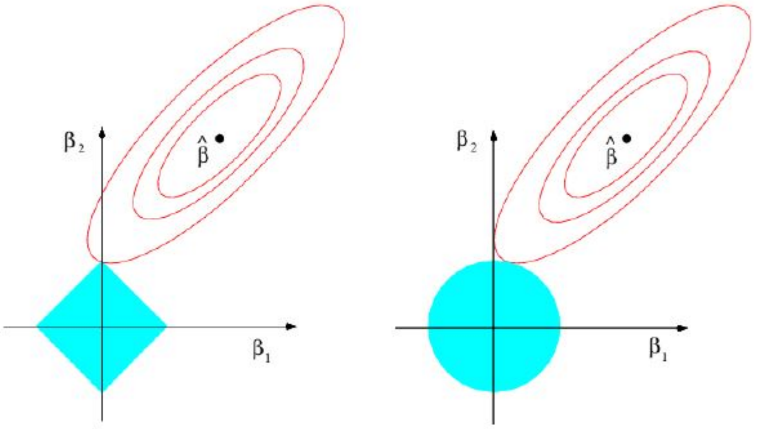


FIGURE 6.7. *Contours of the error and constraint functions for the lasso* (left) *and ridge regression* (right). *The solid blue areas are the constraint regions, $|\beta_1| + |\beta_2| \leq s$ and $\beta_1^2 + \beta_2^2 \leq s$, while the red ellipses are the contours of the RSS.*

(a) As we increase s from 0, the training RSS will:

  i. Increase initially, and then eventually start decreasing in an inverted U shape.

  ii. Decrease initially, and then eventually start increasing in a U shape.

  iii. Steadily increase.

  iv. Steadily decrease.

  v. Remain constant.

The correct answer is iv. and v. As s grows, the coefficients of our predictor variables start approaching the coefficients given by the OLS linear regression, our model is becoming more flexible and RSS decreases. However, once s is large enough (once the diamond contains the minimum point of the contours of RSS in graph 6.7), the training RSS will remain constant and equal the OLS RSS.

(b) Repeat (a) for test RSS. The correct answer is ii. Test RSS depends on the Variance and Bias of our model. When values of s are 0 (or near 0) then the variance of our model is very low, however our bias is very high. When s increases to a point where the diamond in graph 6.7 touches the minimum point of the contour lines of the RSS, our model has very low bias, but extremely high variance. Somewhere in between these two values the test RSS reaches a minimum point. Hence the U shape.

(c) Repeat (a) for variance. The correct answer is iii. As discussed in part b) when s is 0 the variance is low. As s increases the variance increases as well.

(d) Repeat (a) for (squared) bias. The correct answer is iv. As discussed in part b), when s is 0, the bias is extremely high. As s increases the bias decreases.

(e) Repeat (a) for the irreducible error. The correct answer is v. The irreducible error does not depend on s (or anything for that matter). Hence the name irreducible i.e. it stays constant no matter what we do.

5. Suppose we estimate the regression coefficients in a linear regression model by minimizing:

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}{\beta_j}^2$$

for a particular value of λ. For parts (a) through (e), indicate which of i. through v. is correct. Justify your answer.

(a) As we increase λ from 0, the training RSS will:

i. Increase initially, and then eventually start decreasing in an inverted U shape.

ii. Decrease initially, and then eventually start increasing in a U shape.

iii. Steadily increase.

iv. Steadily decrease.

v. Remain constant.

The correct answer is iii. and v. As λ grows, the coefficients of our predictor variables start shrinking to 0, our model is becoming less flexible and RSS increases. However, once λ is large enough, the training RSS will remain constant and equal the null model RSS.

(b) Repeat (a) for test RSS. The correct answer is ii. Test RSS depends on the Variance and Bias of our model. When values of λ are very high (or infinity) then the variance of our model is very low, however our bias is very high, since our model is less flexible. When λ is 0, our model has very low variance, but extremely small bias. Somewhere in between these two values the test RSS reaches a minimum point. Hence the U shape.

(c) Repeat (a) for variance. The correct answer is iv. As discussed in part b) when λ is 0 the variance is high. As λ increases the variance decreases, because our model is becoming less flexible.

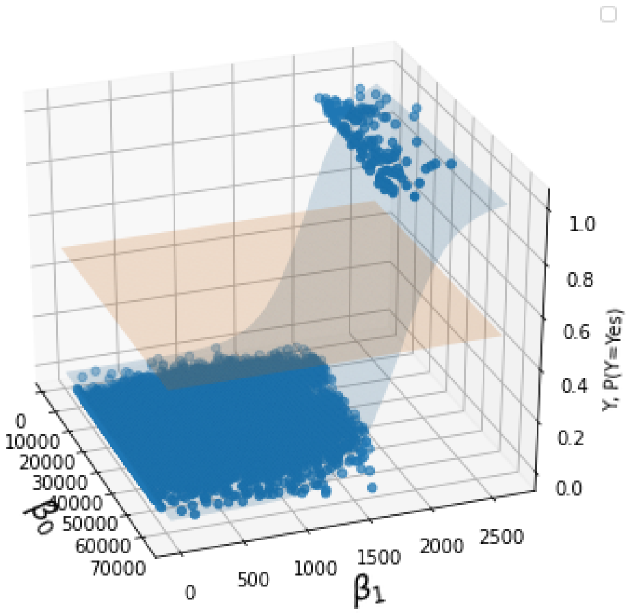
(d) Repeat (a) for (squared) bias. The correct answer is iii. As discussed in part b), when λ is 0, the bias is very low. As λ increases the bias decreases.

(e) Repeat (a) for the irreducible error. The correct answer is v. The irreducible error does not depend on s (or anything for that matter). Hence the name irreducible i.e. it stays constant no matter what we do.

2 Applied Questions

1. We can use the logistic regression to predict the probability of default using income and balance on the Default data set. We will now estimate the test error of this logistic regression model using the validation set approach.

(a) Fit a logistic regression model that uses income and balance to predict default.



The blue dots represent the training data points. The transparent blue plane is the logistic regression fit. The transparent red plane is the 0.5 cutoff decision boundary for the logistic regression.

(b) Using the validation set approach, estimate the test error of this model. In order to do this, you must perform the following steps:

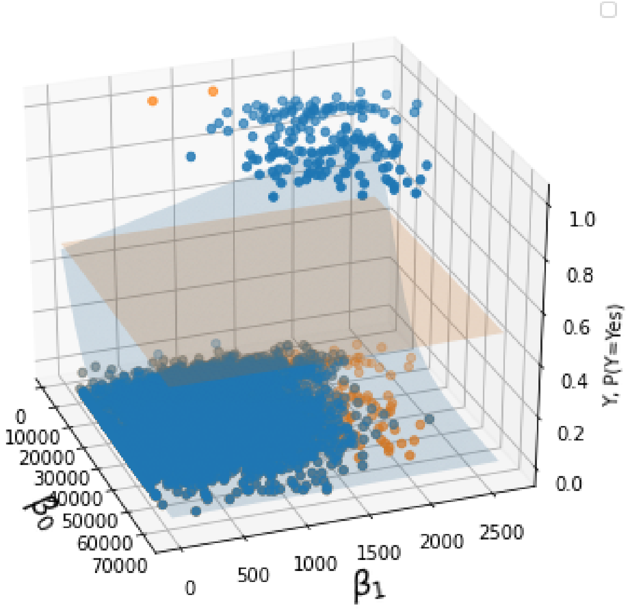
i. Split the sample set into a training set and a validation set.

ii. Fit a multiple logistic regression model using only the training observations.

iii. Obtain a prediction of default status for each individual in the validation set by computing the posterior probability of default for that individual, and classifying the individual to the default category if the posterior probability is greater than 0.5.

iv. Compute the validation set error, which is the fraction of the observations in the validation set that are misclassified.

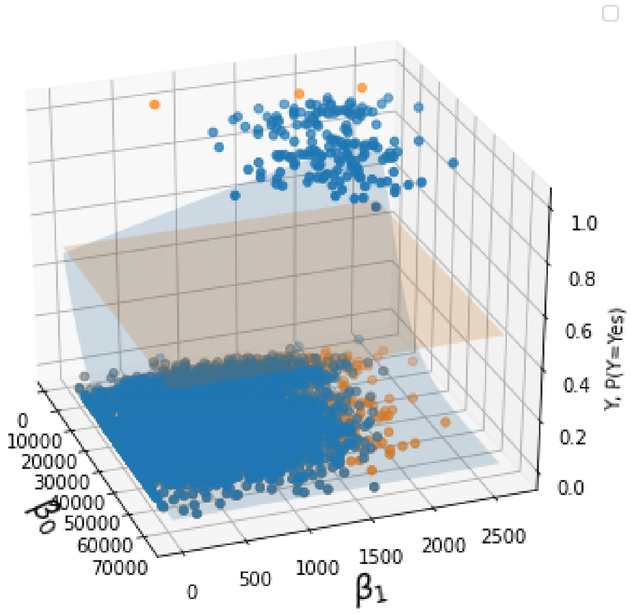
The validation set error was 0.0322 (using a validation set whose size is a half of the original training data). I decided to plot the results:

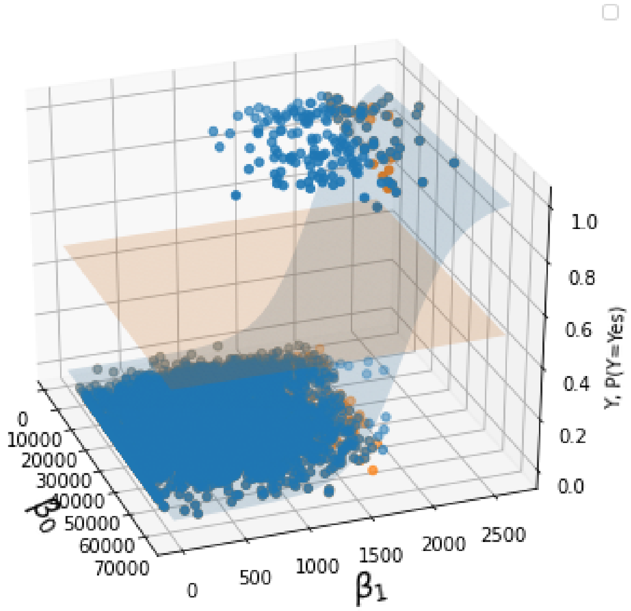


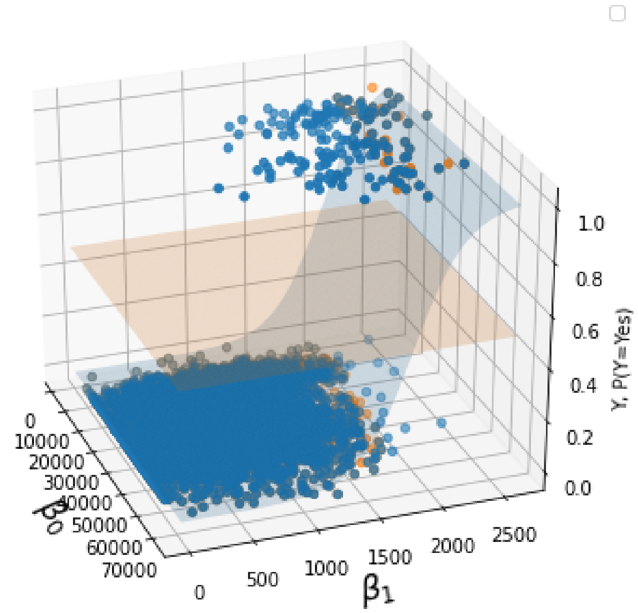
The transparent blue plane is the logistic regression fit. The transparent red plane is the 0.5 cutoff decision boundary for the logistic regression. The blue scatter points are all the points where the prediction and test data match. The orange points are all the points where the prediction and test data do not match.

(c) Repeat the process in (b) three times, using three different splits of the observations into a training set and a validation set. Comment on the results obtained.

The error rates of the three processes were: 0.035, 0.026 and 0.0252 respectively. Below are their respective plots, with same legend as previously:


0
10000
20000
30000
40000
50000
60000
70000
β_0
0
500
1000
1500
2000
2500
β_1
0.0
0.2
0.4
0.6
0.8
1.0
Y, P(Y=Yes)


0
10000
20000
30000
40000
50000
60000
70000
β_0
0
500
1000
1500
2000
2500
β_1
0.0
0.2
0.4
0.6
0.8
1.0
Y, P(Y=Yes)



As we can see, the first of the three model fits is most similar to the model fit in part b), which explains why their error rate is so similar. They are also quite different from the model fit obtained when using all the data from part a). The second and third model fits are similar to each other and also similar to the model fit from part a). This is also the reason why their error rate is so low. It is interesting to see in action the high variance of the test error rates from using the validation set approach, as demonstrated by these 2 widely different types of model fits.

(d) Now consider a logistic regression model that predicts the probability of default using income, balance, and a dummy variable for student. Estimate the test error for this model using the validation set approach. Comment on whether or not including a dummy variable for student leads to a reduction in the test error rate.

Using the same validation set from part b), I managed to obtain a test error rate of 0.0288, which is a 3.4% reduction.

2. We will now perform cross-validation on a simulated data set.

(a) Generate a simulated data set as follows:
- x: create 100 random samples from normal distribution with mean 0 and variance 1
- $y = x - 2x^2 + noise$
noise are samples from normal distribution with mean 0 and variance 1

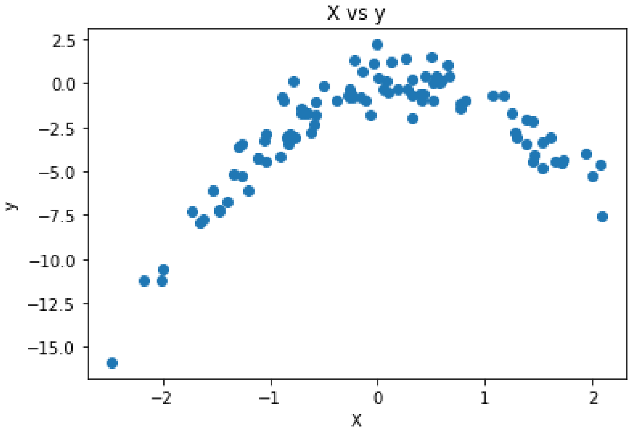
In this data set, what is n and what is p? Write out the model used to generate the data in equation form.
$n = 100$ since we have a sample of size 100, and since we have 2 predictors, X and X^2, $p = 2$

$$X \sim N(0,1)$$

$$\epsilon \sim N(0,1)$$

$$Y = X - 2X^2 + \epsilon$$

(b) Create a scatterplot of X against Y . Comment on what you find.



As we can see the scatter plot has the expected shape of an inverted parabola centered at $X = 0$. We see that most data points occur within the [-2,2] interval, which is the 95% confidence interval. We see no extremely high leverage points, since no data points deviate more than 3 standard deviations from $X = 0$. There is a slight skew of moderately high leverage points to the left of $X = 0$

(c) Compute the LOOCV errors that result from fitting the following four models using least squares:

i. $Y = \beta_0 + \beta_1 X + \epsilon$

ii. $Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \epsilon$

iii. $Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \epsilon$

iv. $Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \beta_4 X^4 + \epsilon$

For this part I have used formula (5.2) which states that the cross-validation error in a OLS linear or polynomial fit is:

$$CV_{(n)} = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{y_i - \hat{y}_i}{1 - h_i} \right)^2$$

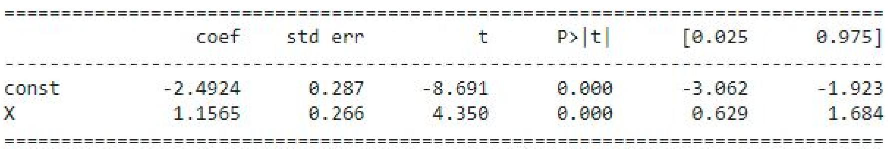
Linear Fit Error = 8.18736992
Quadratic Fit Error = 0.89812155
Cubic Fit Error = 0.89275131
Quartic Fit Error = 0.86450403

Here we see a big improvement in the cross-validation error when going from a linear model to a quadratic model (which makes sense since the true relationship is quadratic). The cross-validation error of the quadratic, cubic and quartic fit is relatively the same, implying that the cubic and quartic fit are likely overfitting the data, which the quadratic fit can model just as well.

(d) Which of the models in (c) had the smallest LOOCV error? Is this what you expected? Explain your answer. The smallest error was found with the quartic fit (i.e. polynomial of fourth degree). Although it is smallest, it is a relatively low improvement over the quadratic fit. This does not come as a surprise, because when using LOOCV, the LOOCV error is expected to level off at a the lowest flexibility level which is able to capture the true relationship of a model. In this case this is the quadratic model.

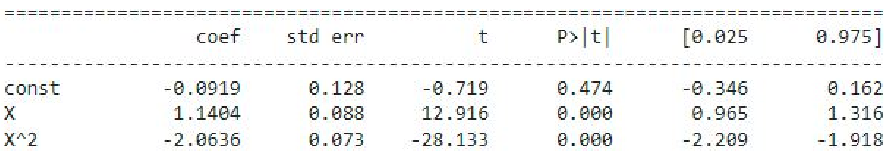
(e) Comment on the statistical significance of the coefficient estimates that results from fitting each of the models in (c) using least squares. Do these results agree with the conclusions drawn based on the cross-validation results?

Linear Regression:



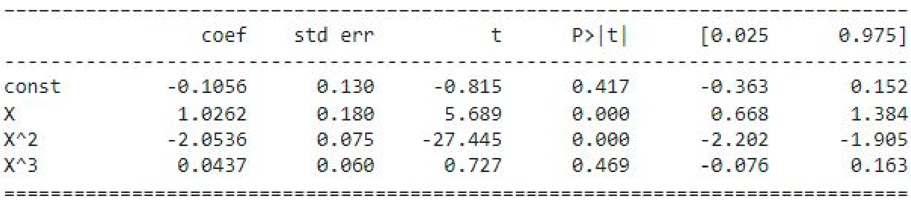
```
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         -2.4924      0.287     -8.691      0.000      -3.062      -1.923
X              1.1565      0.266      4.350      0.000       0.629       1.684
==============================================================================
```

Given that the true relationship is not linear, we expect high statistical significance of both the β_0 and β_1 in a linear fit, since both of these are likely non-zero (since a line is trying to approximate a parabola). Quadratic Regression:



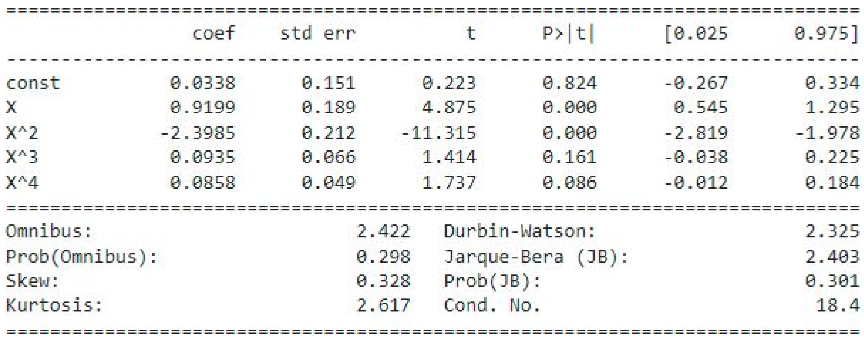
```
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.0919      0.128     -0.719      0.474      -0.346       0.162
X              1.1404      0.088     12.916      0.000       0.965       1.316
X^2           -2.0636      0.073    -28.133      0.000      -2.209      -1.918
==============================================================================
```

Here the only term with no statistical significance is the constant term. Both, β_1 and β_2 are statistically significant. Together all three values are extremely close to their values in the true relationship (0, 1 and -2 respectively). In relation to part c), this introduction of a statistically significant higher order term corresponds to a significant decrease in LOOCV test error (relative to the linear model fit). Cubic Regression:



```
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.1056      0.130     -0.815      0.417      -0.363       0.152
X              1.0262      0.180      5.689      0.000       0.668       1.384
X^2           -2.0536      0.075    -27.445      0.000      -2.202      -1.905
X^3            0.0437      0.060      0.727      0.469      -0.076       0.163
==============================================================================
```

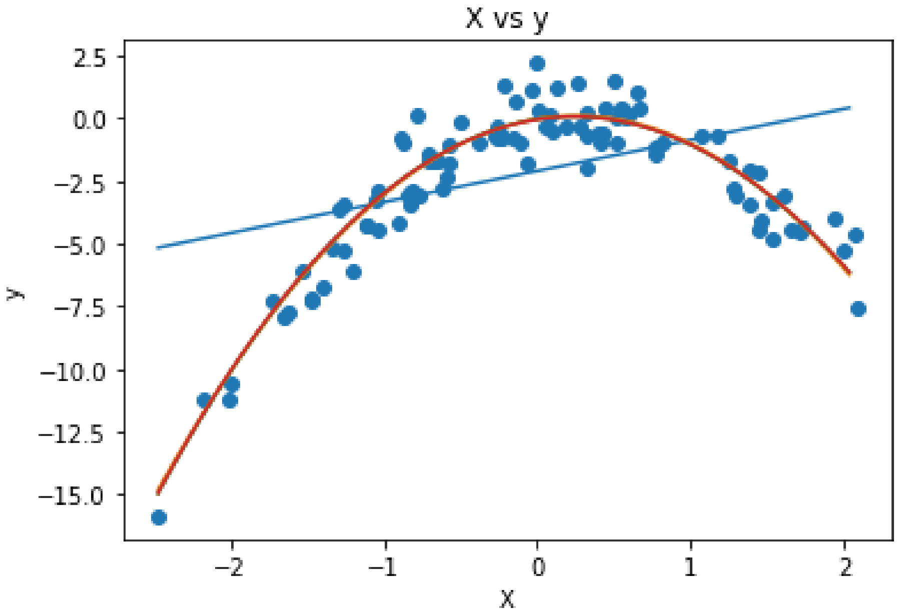
Here is where things get interesting again. The terms now with no statistical significance are β_0 and β_3, implying that we are likely overfitting our data by using a cubic term. In relation to part c), this introduction of a statistically insignificant higher order term corresponds to an insignificant decrease in LOOCV test error (relative to the quadratic model fit). Quartic Regression:



```
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0338      0.151      0.223      0.824      -0.267       0.334
X              0.9199      0.189      4.875      0.000       0.545       1.295
X^2           -2.3985      0.212    -11.315      0.000      -2.819      -1.978
X^3            0.0935      0.066      1.414      0.161      -0.038       0.225
X^4            0.0858      0.049      1.737      0.086      -0.012       0.184
==============================================================================
Omnibus:                        2.422   Durbin-Watson:                   2.325
Prob(Omnibus):                  0.298   Jarque-Bera (JB):                2.403
Skew:                           0.328   Prob(JB):                        0.301
Kurtosis:                       2.617   Cond. No.                         18.4
==============================================================================
```

Here the terms with no statistical significance are β_0, β_3 and β_4, implying that we are again likely overfitting our data by using a quartic term. In relation to part c), this introduction of a statistically insignificant higher order term corresponds to an insignificant decrease in LOOCV test error (relative to the quadratic and cubic model fit).

Here is a plot of all 4 model fits:



Here, we see that the quadratic, cubic and quartic fit are literally overlapping each other, which implies that using cubic and quartic terms is an overkill i.e. overfitting.

3. We will now consider the Boston housing data set.

(a) Based on this data set, provide an estimate for the population mean of medv. Call this estimate $\hat{\mu}$.
An estimate for the population mean is the sample mean, which is: 22.532806324110698

(b) Provide an estimate of the standard error of $\hat{\mu}$. Interpret this result.
An estimate for the standard deviation of the sample mean is the standard deviation of the sample mean, divided by the square root of sample size i.e.:

$$\hat{\sigma} = \frac{\sigma}{\sqrt{n}} = 0.4088611474975351$$

Suppose we take many different samples of size n from a population, and calculate the mean of these samples. Once, done we compute the mean of all the sample means. $\hat{\sigma}$ represents an estimate of the standard deviation of the mean value of sample means.

(c) Now estimate the standard error of $\hat{\mu}$ using the bootstrap. How does this compare to your answer from (b)? Using the bootstrap method, I have obtained:

$$\hat{\sigma} = 0.407870612804229$$

This value is incredibly close to the estimate of b).

(d) Based on your bootstrap estimate from (c), provide a 95% confidence interval for the mean of medv. The 95% confidence interval is approximately $[\hat{\mu} - 2\hat{\sigma}\ ,\ \hat{\mu} + 2\hat{\sigma}] = [21.71706509850224, 23.348547549719157]$

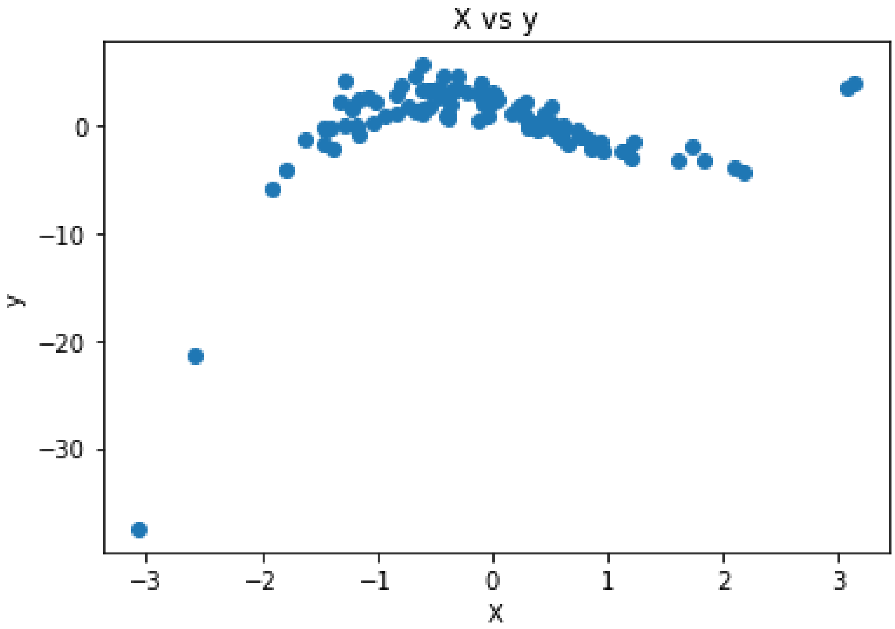
4. Here, we will generate simulated data, and will then use this data to perform best subset selection.

(a) Generate a predictor X of length n = 100 from a normal distribution with mean 0 and variance 1, as well as a noise vector ϵ of length n = 100.

(b) Generate a response vector Y of length n = 100 according to the model

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \epsilon$$

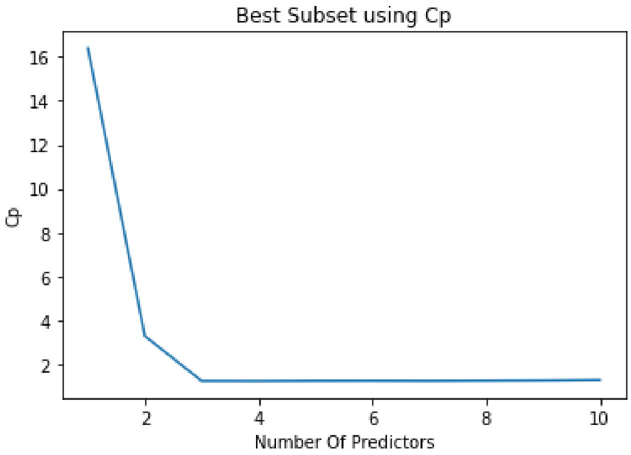
where β_0, β_1, β_2, β_3 are constants of your choice. For X and ϵ use the data being generated in (a).



With $\beta_0 = 2$, $\beta_1 = -3$, $\beta_2 = -2$, $\beta_3 = 1$

(c) Perform best subset selection in order to choose the best model containing the predictors X, X^2, ... , X^{10}. What is the best model obtained according to C_p, BIC, and adjusted R^2? Show some plots to provide evidence for your answer, and report the coefficients of the best model obtained.

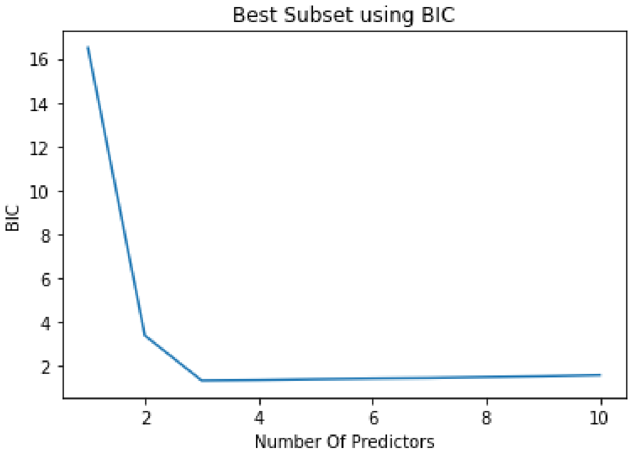
Here is a graph of Cp values against k number of predictors used (where the predictors are the predictors which give the largest R^2, for a given k).



The best model is the one with the lowest Cp score, and in this case it is the one with 4 predictors, namely (X, X^2, X^3 and X^4). The coefficients are: $\beta_0 = 1.9154183085793952$, $\beta_1 = -3.00239904$, $\beta_2 = -1.77720647$, $\beta_3 = 1.02590593$, $\beta_4 = -0.02854208$

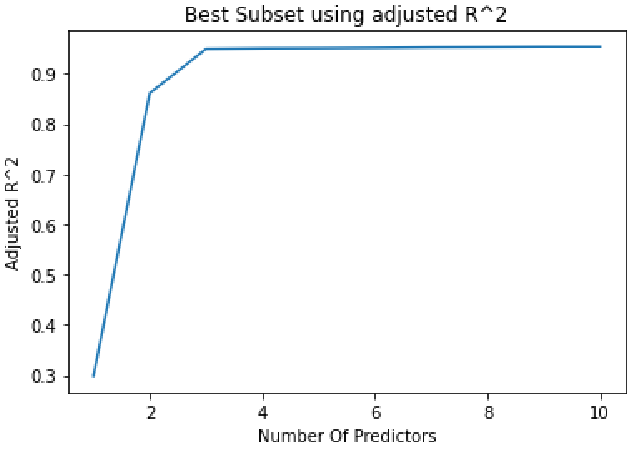
Note: Cp and BIC need an unbiased estimate for the standard deviation of the error term. Although we know (because data is simulated) that the variance of the error term is 1, it is still proper to estimate it manually. This was done according to this thread on stack exchange. In other words $\sigma = \sqrt{\frac{RSS}{n-p}}$.

Here is a graph of BIC values against k number of predictors used:



You would be correct to say that this graph looks very similar to that of Cp. BIC assigns a slightly larger penalty for a greater number of predictors. The best model is the one with the lowest BIC score, and in this case it is the one with 3 predictors, namely (X, X^2 and X^3). The coefficients are: $\beta_0 = 2.037227598621563$, $\beta_1 = -3.01703668$, $\beta_2 = -2.00523335$, $\beta_3 = 1.02136414$

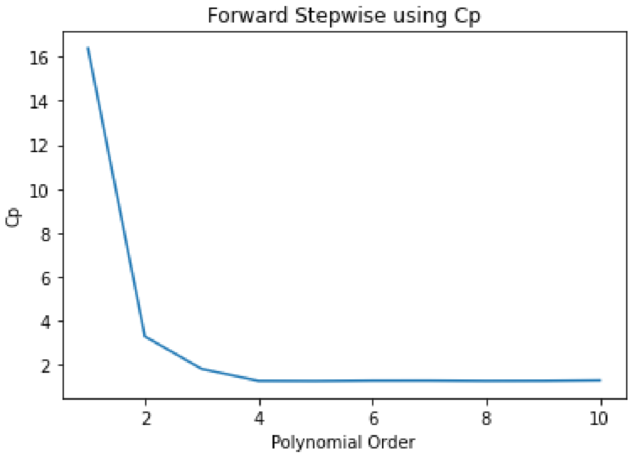
Finally, here is a graph of adjusted R^2 values against k number of predictors used:



The graph looks different because adjusted R^2 assigns a higher value to models which are a better fit, but which also do not have too many predictors. However, it seems as though this cost of having too many predictors was not high enough, since the selection process suggest that the model using all 10 powers of X is the best one. The coefficients are: $\beta_0 = 2.1939089512687406$, $\beta_1 = -3.08995163e+00$, $\beta_2 = -3.74523556e+00$, $\beta_3 = 1.28936699e+00$, $\beta_4 = 2.06788457e+00$, $\beta_5 = -7.43233306e-02$, $\beta_6 = -6.96725942e-01$, $\beta_7 = 6.63999960e-04$, $\beta_8 = 8.80098151e-02$, $\beta_9 = 4.70264447e-04$, $\beta_{10} = -3.75301283e-03$

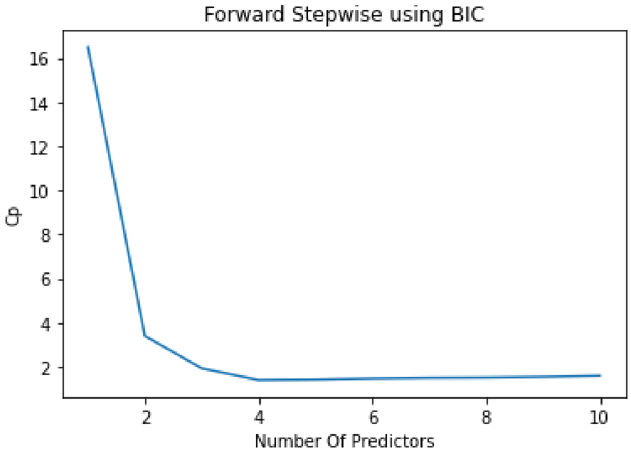
(d) Repeat (c), using forward stepwise selection and also using backwards stepwise selection. How does your answer compare to the results in (c)? Forward Stepwise Selection:

Using Cp the following graph was obtained:



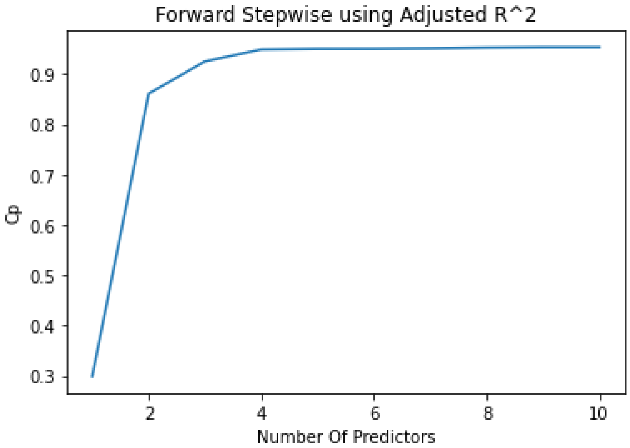
Here the best model is one using 5 predictors: X, X^2, X^3, X^4 and X^5. The coefficients are: $\beta_0 = 1.9124384716086256$, $\beta_1 = -2.90069485$, $\beta_2 = -1.76702905$, $\beta_3 = 0.95330166$, $\beta_4 = -0.03046272$, $\beta_5 = 0.00696324$. Compared to best subset selection using Cp in c), this model uses just one more predictor, has similar values for its coefficients, and the coefficient for the extra predictor is extremely small.

Using BIC the following graph was obtained:



Here the best model is one using 4 predictors: X, X^2, X^3 and X^5. The coefficients are: $\beta_0 = 2.039372365676964$, $\beta_1 = -2.97562905e + 00$, $\beta_2 = -2.00735755e + 00$, $\beta_3 = 9.91389463e - 01$, $\beta_5 = 2.86272407e - 03$. Compared to best subset selection using BIC in c), this model uses just one more predictor, has similar values for its coefficients, and the coefficient for the extra predictor is extremely small.

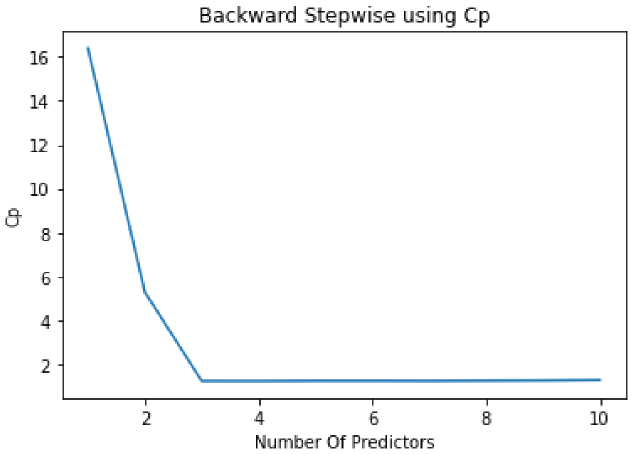
Using adjusted R^2 the following graph was obtained:



Here the best model is one using all predictors. The coefficients are: $\beta_0 = 2.1939089512687406$, $\beta_1 = -3.08995163e+00$, $\beta_2 = -3.74523556e+00$, $\beta_3 = 1.28936699e+00$, $\beta_4 = 2.06788457e+00$, $\beta_5 = -7.43233306e-02$, $\beta_6 = -6.96725942e-01$, $\beta_7 = 6.63999960e-04$, $\beta_8 = 8.80098151e-02$, $\beta_9 = 4.70264447e-04$, $\beta_{10} = -3.75301283e-03$. Compared to best subset selection using adjusted R^2 in c), this model uses the same predictors, and has the same values for its coefficients.

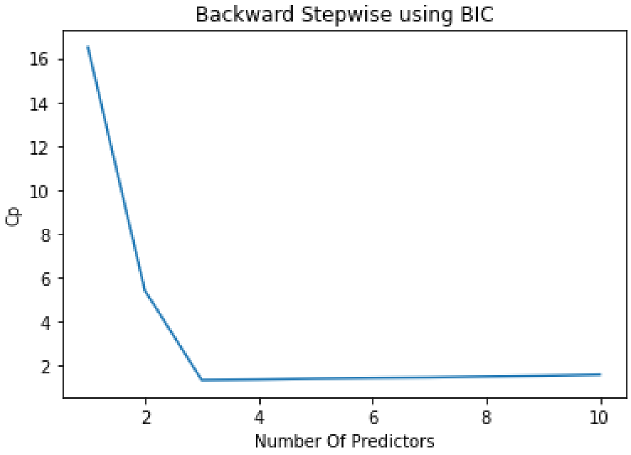
Backward Stepwise Selection:

Using Cp the following graph was obtained:



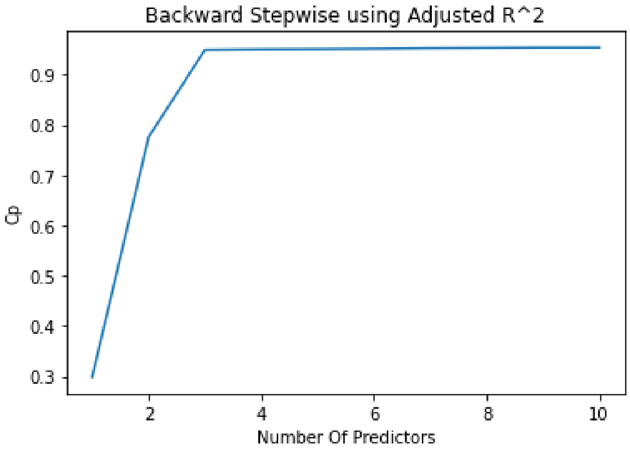
Here the best model is one using 4 predictors: X, X^2, X^3 and X^6. The coefficients are: $\beta_0 = 1.9400523014177835$, $\beta_1 = -3.01315533e + 00$, $\beta_2 = -1.85546596e + 00$, $\beta_3 = 1.02807143e + 00$, $\beta_6 = -2.17369276e - 03$. Compared to best subset selection using Cp in c), this model uses the same number of predictors, although in c) we had a X^4 term, and here we have a X^6. The values of the corresponding coefficients are also very similar. The main difference are coefficients β_4 and β_6, the former being small and the latter being extremely small.

Using BIC the following graph was obtained:



Here the best model is one using 3 predictors: X, X^2 and X^3. The coefficients are: $\beta_0 = 2.037227598621563$, $\beta_1 = -3.01703668$, $\beta_2 = -2.00523335$, $\beta_3 = 1.02136414$. Compared to best subset selection using BIC in c), this model uses the same predictors, and has the same values for its coefficients.

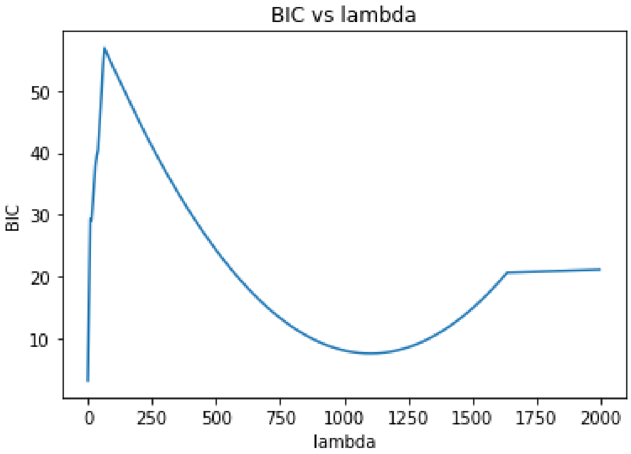
Using adjusted R^2 the following graph was obtained:



Here the best model is one using all predictors. The coefficients are: $\beta_0 = 2.1939089512687406$, $\beta_1 = -3.08995163e+00$, $\beta_2 = -3.74523556e+00$, $\beta_3 = 1.28936699e+00$, $\beta_4 = 2.06788457e+00$, $\beta_5 = -7.43233306e-02$, $\beta_6 = -6.96725942e-01$, $\beta_7 = 6.63999960e-04$, $\beta_8 = 8.80098151e-02$, $\beta_9 = 4.70264447e{-}04$, $\beta_{10} = -3.75301283e{-}03$. Compared to best subset selection using adjusted R^2 in c), this model uses the same predictors, and has the same values for its coefficients.

(e) Now fit a lasso model to the simulated data, again using X, X^2, ... , X^{10} as predictors. Use cross-validation to select the optimal value of λ. Create plots of the cross-validation error as a function of λ. Report the resulting coefficient estimates, and discuss the results obtained.

Here is the graph of the CV-score against lambda:

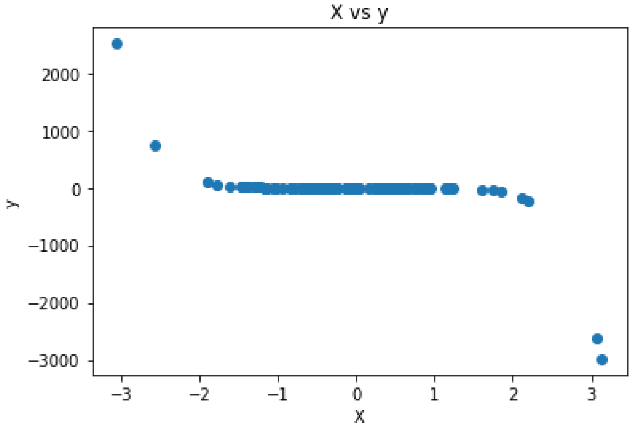


The CV score is the average BIC score of k folds. Surprisingly the Lasso regression seems to prefer lambda to be 0, in which case all polynomial terms would be included and the coefficients are: $\beta_0 = 1.8845692976141497$, $\beta_1 = -2.92785059e+00$, $\beta_2 = -1.71518261e+00$, $\beta_3 = 1.03177182e+00$, $\beta_4 = -5.06691823e-03$, $\beta_5 = -1.99096579e-02$, $\beta_6 = -1.15187991e-02$, $\beta_7 = 2.02912376e - 03$, $\beta_8 = 1.92684933e - 04$, $\beta_9 = -1.23416900e - 05$, $\beta_{10} = 7.18647635e - 05$.
Curiously, these are not the same as in linear regression, but after some research I have found that sklearns Lasso regression does not converge well for lambda=0.
I have wondered why lasso made such a poor choice, and after playing a bit with lambda, I have found that Lasso tends to drastically decrease β_0, β_1, β_2 and β_3 as lambda increases from 0 to 1.

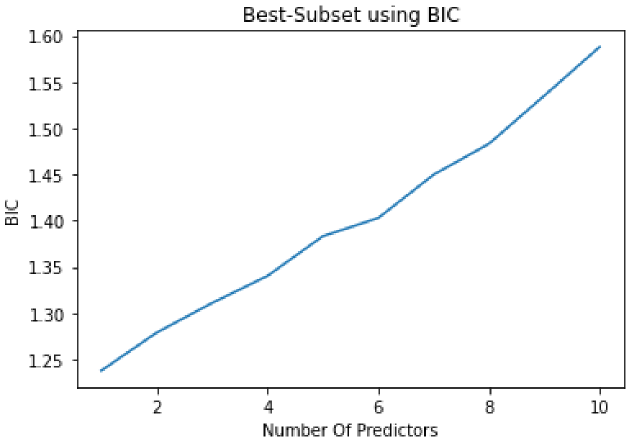
(f) Now generate a response vector Y according to the model

$$Y = \beta_0 + \beta_7 X^7 + \epsilon$$

and perform best subset selection and the lasso. Discuss the results obtained.
The parameters chosen are $\beta_0 = 2$ and $\beta_7 = -1$. Here is a scatter plot of the data:

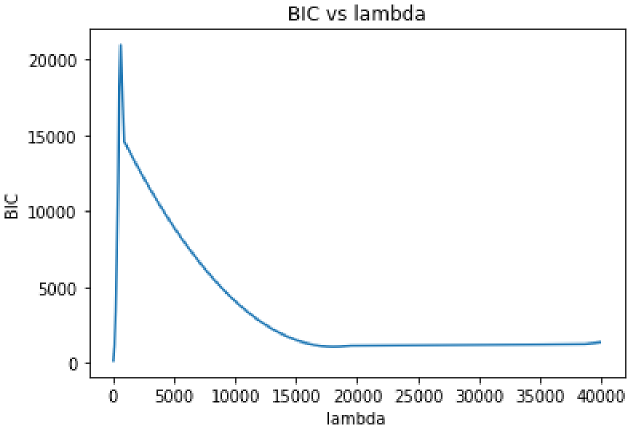


Using best subset selection and BIC as the CV scoring system, I obtained the following BIC vs number of predictors graph:

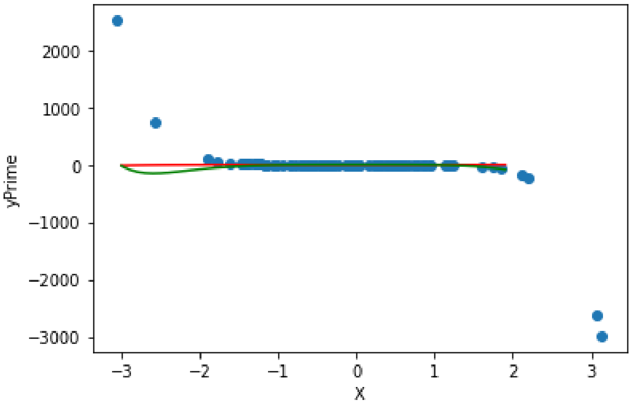


This isn't too surprising, as the model with only 1 predictor is expected to perform best. The selection system appropriately identified X^7 to be the best predictor, however what is surprising is that are estimates of parameters: $\beta_0 = -0.008870002240835062$ and $\beta_7 =, 0.00569308$.

Similarly, using the BIC as the CV scoring system for determining the best lambda, the following graph was obtained:



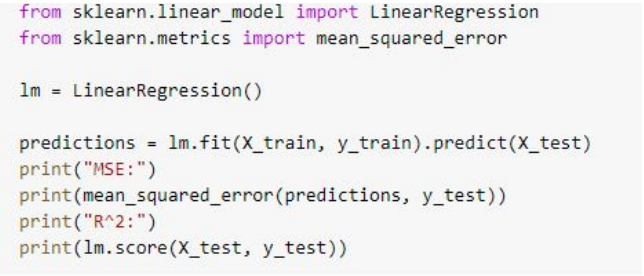
Here, again lasso seems to prefer when lambda is zero. The coefficients are: $\beta_0 = 1.6179938580208244$, $\beta_1 = -6.79501531e{+}00$, $\beta_2 = 6.85667462e{-}01$, $\beta_3 = 1.15347087e + 01$, $\beta_4 = 6.52983485e - 01$, $\beta_5 = -4.54386338e + 00$, $\beta_6 = -1.94350370e - 01$, $\beta_7 = -3.83595052e - 01$, $\beta_8 = 1.58513519e - 03$, $\beta_9 = -2.71812133e - 02$, $\beta_{10} = 1.13954867e - 03$.
I decided to plot both fits:



Now it is a bit more clear why the fits cannot really find the correct parameters. The values which give the true relationship its characteristic shape are outliers and leverage points (i.e. points which are few in number), and the models (in order not to overfit the data) do not fit them.

5. Here, we will predict the number of applications received using the other variables in the College data set.

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



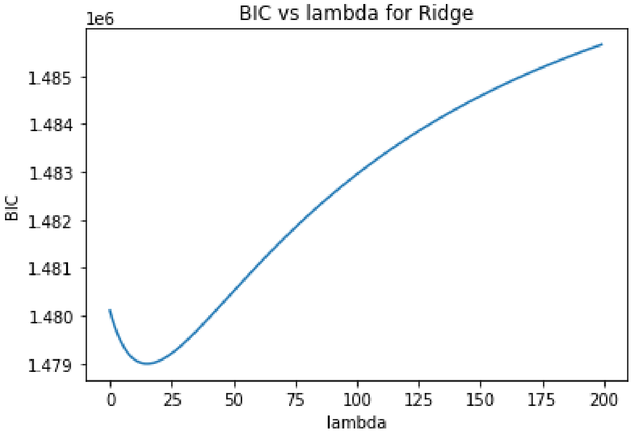
```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

lm = LinearRegression()

predictions = lm.fit(X_train, y_train).predict(X_test)
print("MSE:")
print(mean_squared_error(predictions, y_test))
print("R^2:")
print(lm.score(X_test, y_test))
```

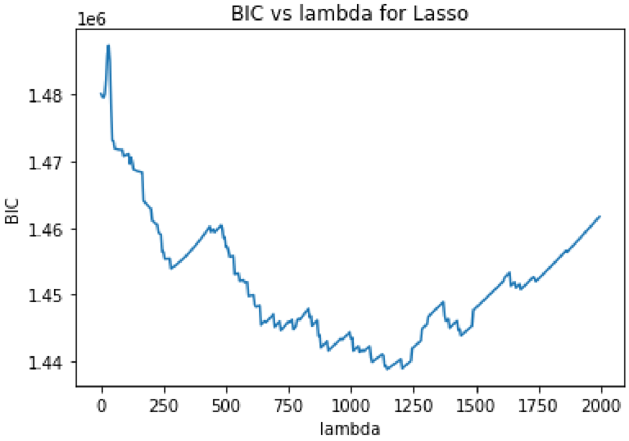
(b) Fit a linear model using least squares on the training set, and report the test error obtained.
The test $MSE = 1654196.509145673$
The test $R^2 = 0.9113754682885489$

(c) Fit a ridge regression model on the training set, with λ chosen by cross-validation. Report the test error obtained.
Here is a BIC vs. lambda graph:



Using lambda with smallest BIC ($\lambda = 15.0$):
Test $MSE = 1654196.509145673$
Test $R^2 = 0.9115284414037086$

(d) Fit a lasso model on the training set, with λ chosen by cross-validation. Report the test error obtained, along with the number of non-zero coefficient estimates. Here is a BIC vs. lambda graph:



Using lambda with smallest BIC ($\lambda = 1145$):
Test $MSE = 1654196.509145673$
Test $R^2 = 0.9114603470832264$
The number of non-zero coefficient estimates is 17.