1. Model 1 and Model 2 are two linear regression models. Suppose features of Model 1 are a strict subset of those in Model 2. Which **two** of the following statements are true?

Here, training/test error = squared error on training/test dataset. Further to rule out some trivial cases (e.g., both models fit data perfectly and have zero training error), let's assume the training/test errors from the two models are different.

1 point

* It is impossible to tell which model has a smaller test error with the given information.
* ~~Model 1 has a smaller test error.~~
* ~~Model 1 has a smaller training error.~~
* ~~Model 2 has a smaller test error.~~
* ~~It is impossible to tell which model has a smaller training error with the given information.~~
* Model 2 has a smaller training error.

Which **two** of the following statements are true, where performance refers to the averaged squared error?

1 point

* ~~Adding more features to the model helps prevent overfitting on the~~ **~~training~~** ~~data~~
* For OLS, introducing more features always results in equal or better performance on the **test** data
* For OLS, introducing more features always results in equal or better performance on the **training** data
* Adding more features to the model makes it more likely to overfit the **training** data.

1. Which statement is true, where performance refers to the averaged squared error?

1 point

* Introducing ridge or Lasso regularization to the model always results in equal or worse performance on the **training** data than the one produced by OLS
* Introducing ridge or Lasso regularization to the model always results in equal or worse performance on the **test** data than the one produced by OLS
* Introducing ridge or Lasso regularization to the model always results in equal or better performance on the **test** data than the one produced by OLS
* Introducing ridge or Lasso regularization to the model always results in equal or better performance on the **training** data than the one produced by OLS

Question 4

Which **two** of the following statements about regularization (used in Lasso) are true

* If a feature's coefficient is shrunk to zero with *λ*1​, then this feature will stay zero for any *λ* greater than *λ*1​.
* If a feature's coefficient is shrunk to zero with *λ*1​, then this feature will stay zero for any *λ* less than *λ*1​.
* Before applying Lasso, we should center and scale the features to make the features look like random samples from a N(0, 1)*N*(0,1) distribution.
* A large value of *λ* can cause your model to underfit the data
* *λ*min​ is always smaller (or at least not bigger) than *λ*1se​((a (
* A large value of *λ* can cause your model to overfit the data

Question 5

Which of the following statements about regularization (used in Lasso) is true?

* Let *βj*​(*λ*) denote the coefficient for feature j*j* returned by Lasso using regularization parameter λ. Suppose we have a total of *p* features. Then when we increase λ, ∑*j*=1*p*​*βj*​(*λ*)2, i.e., the norm square of the *p*-dimensional coefficient vector *β*(*λ*), is decreasing (or not increasing).
* Let \beta\_j(\lambda)*βj*​(*λ*) denote the coefficient for feature j*j* returned by Lasso using regularization parameter λ. Then when we increase λ, the absolute value of \beta\_j(\lambda)*βj*​(*λ*) is decreasing (or not increasing).
* When we increase *λ*, the size of the model selected by Lasso is increasing (or at least not decreasing).
* When we increase *λ*, the size of the model selected by Lasso is decreasing (or at least not increasing). (This also seems to be correct in case we had also used cross validation technique)
* Let *βj*​(*λ*) denote the coefficient for feature j*j* returned by Lasso using regularization parameter *λ*. Suppose we have a total of *p* features. Then when we increase *λ*,∑*j*=1*p*​∣*βj*​(*λ*)∣ is decreasing (or not increasing.
* Let \beta\_j(\lambda)*βj*​(*λ*) denote the coefficient for feature j*j* returned by Lasso using regularization parameter λ. Then when we increase λ, the absolute value of \beta\_j(\lambda)*βj*​(*λ*) is increasing (or not decreasing).

6. In ridge regression, a large *λ* value leads to a model with

(Hint: there are two correct answers.)

* low variance
* high variance
* high bias
* low bias

*The large lambda is intended to reduce the variance at the cost of increasing the Bias Slightly*

7. Suppose we fit a ridge regression model with a fixed lambda value and option "standardize = FALSE" in R

glmnet(X, Y, alpha = 0, lambda = 0.5, standardize = FALSE)

If we double the value of the first feature (i.e., multiply the 1st column of X by two), then use the command above to refit a ridge regression model.

X[,1] = 2 \* X[,1]

glmnet(X, Y, alpha = 0, lambda = 0.5, standardize = FALSE)

What would happen to the estimated coefficients of **the** **other features**? They:

1 point

* Double
* Stay the same
* Half
* Impossible to tell from the information provided

8. Suppose we fit a ridge regression model with a fixed lambda value. Now we choose option "standardize = TRUE"

glmnet(X, Y, alpha = 0, lambda = 0.5, standardize = TRUE)

If we double the value of the first feature (i.e., multiply the first column of X by two), then use the command above to refit a ridge regression model.

X[,1] = 2 \* X[,1]

glmnet(X, Y, alpha = 0, lambda = 0.5, standardize = TRUE)

* Double
* Stay the same (Chose this in first and now in second. Hope this is right since that’s what my calculation shows)
* Half
* Impossible to tell from the information provided

9. Continue with Question 8. What would happen to the estimated coefficient of **the first feature**?

* Double
* Stay the same
* Half
* Impossible to tell from the information provided

10. In subset selection using AIC, if we double the value of a feature (i.e., multiply a specific column of the design matrix by two), what would happen to the estimated coefficients of **the other features**? They:

**Note:** If a feature is not chosen by AIC, its estimated coefficient is zero.

1 point

* Stay the same
* Half
* Double
* Impossible to tell from the information provided

11. The remaining questions are related to the "prostate'' data. Load the data and split the data into training and test by the variable "train". The training set should contain 67 samples and the test contains 30 samples.

prostate = read.table("https://web.stanford.edu/~hastie/ElemStatLearn/datasets/prostate.data",

header = TRUE)

names(prostate)

traindata = prostate[prostate$train==TRUE,]

testdata = prostate[prostate$train==FALSE,]

dim(traindata)

dim(testdata)

All the regression algorithms (mentioned in the remaining questions) should be applied on the **training** set and evaluate the prediction error on the **test** set (i.e., the sum of squared difference between the observed and the prediction over the 30 test samples).

Fit a linear regression model to predict "lpsa'' using the other 8 features (of course, exclude the feature "train''). What's the test error?

**Round your answer to the second decimal place.**

15.64

12.

Continue with the "prostate'' data. Use AIC and BIC to select the optimal model. Since the total number of features is small, you should use exhaustive search, instead of using any stepwise/forward/backward search algorithm. That is, use the R command " **regsubsets** '', instead of " **step** ''.

The optimal model selected by **AIC** contains \_\_\_\_\_\_\_ (an integer, e.g., 4 or 5) non-intercept features.

7

13. The optimal model selected by **BIC** contains \_\_\_\_\_\_\_ (an integer, e.g., 4 or 5) non-intercept features.

2

14. The prediction error on the test data from the model selected by **AIC** is **smaller** or **larger** than the error from the model selected by **BIC**?

* smaller
* larger

15. Continue with the "prostate'' data. Now use Lasso with option "standardize = TRUE" and compute the prediction error. Try the following three lambda values,

* Model 1: lambda = 0.5;
* Model 2: lambda = 0.1;
* Model 3: lambda = 0.01.

Which model has the smallest prediction error?

1 point

Model 1: lambda = 0.5

Model 2: lambda = 0.1

Model 3: lambda = 0.01