

mODEL SELECTION

INTERMEDATE ANALYTICS



June 13, 2018

vICTORIA PLANGE

ALY 6015 SEC 03

SPRING 2018 ,ASSIGNMENT 3

PROFESSOR JUSTIN RODGERS

Regression analysis is used to determine which of the independent variables in our data are related to dependent variables and to understand what entails in those relationships.

For predictive analytics, we build a predictive model with training data, fit the model on it and use the model estimates to determine predicted values of Y. In machine learning, models are trained on subsets of the data. We then calculate the results of each training sample from the training data. All this is done to measure the accuracy of the prediction by trying to reduce bias and variance. We also want to reduce error to increase accuracy and validity. Thus, we assess prediction accuracy by eliminating variables that seem to affect the performance of our model on our training data set. This is known as feature reduction. There are also other methods such as regularization which don’t require the elimination of variables, but rather reduces our parameter estimates (coefficients). The regularization parameter (lambda), would penalize all parameters with the exception of the intercept, so the model is not overfitted.

Question 1

In this assignment, we will explore model selection and regularization with a given data set on weather attributes. We first split the data into a training data set and a test data set. This is done because it is impossible to acquire future data, so we often take a sample from our original data and treat it as though it was future data. This is used to develop the predictive model, and the other is used to assess our current model’s performance. 75% of the data randomly goes to the training data set and 25% goes to the test data set. We set the seed as (12345) to allow us to obtain the same results when the model is repeated.

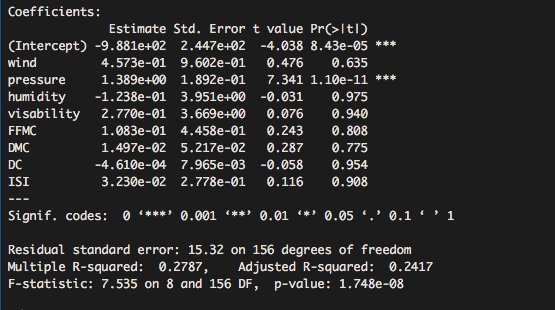
set.seed(12345)

data3\_split <- sample.split(data3, SplitRatio =0.75)

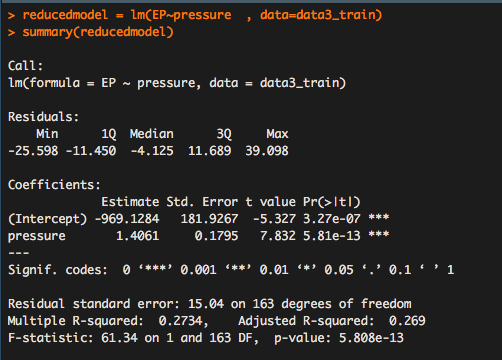
data3\_train <- subset(data3, data3\_split==TRUE) # training data

data3\_test <- subset(data3, data3\_split==FALSE) #testing data

We employ manual feature reduction by using linear regression, to allow us to predict Energy Production (EP), a variable in the data set. We first begin with the saturated model which has all the X variables and perform linear regression on it.



The adjusted R^2(how close the data is from the regression line), is 0.2417 and the f-statistic is 7.535. This means that the 24% of variations in the dependent variable is considered by variations in the independent variable. Next, we refit the model based on only significant variables, which in this case is ‘pressure’.

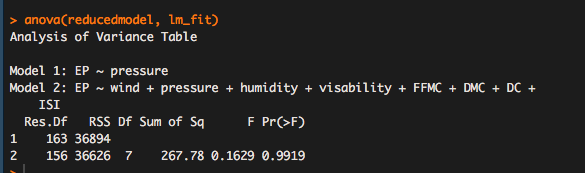


The adjusted R^2 is 0.269 and was a slight nominal increase from the saturated model. When we eliminated certain variables one at a time, we noticed that they had an effect on the adjusted r^2. For instance, visability reduced r^2 when we eliminated it first. The model thus is not changing significantly. The data is insufficient and there may be other missing variables which are not used in the model and thus affecting EP. We conclude that only pressure is important for this model.

To check further if the saturated model is different from the reduced model, we perform anova test:

#Anova

anova(reducedmodel, lm\_fit)



The null hypothesis here is that the null model is our actual model.

With an F-statistic P-value of 0.9919 we fail to reject the null hypothesis because the p value is higher than 0.05. Thus, the model with only the variable ‘pressure’, is the better model.

The models can also be manually written out as follows:

Saturated model: Y = β0 + β1X1 + β2X2 + ε

= -9.881e+02 + 4.573e-01(wind) +1.389+00(pressure) + -1.238e-01(humidity) + 1.083e-01(FFMC) +2.770e-01(visability)

1.497e-02(DMC) + -4.1610e-04(DC) + 3.230e-02 + 15.32

Reduced model: Y = β0 + β1X1 + ε

= -969.1284 + 1.4061 + 15.04

Saturated Model

|  |
| --- |
| Coefficients: |
| Estimate Std.Error t-value Pr(>|t|) |
| (Intercept) -9.881e+02 2.447e+02 -4.038 8.43e-05 \*\*\* |
| wind 4.573e-01 9.602e-01 0.476 0.635 |
| pressure 1.389e+00 1.892e-01 7.341 1.10e-11 \*\*\* |
| humidity -1.238e-01 3.951e+00 -0.031 0.975 |
| visability 2.770e-01 3.669e+00 0.076 0.940 |
| FFMC 1.083e-01 4.458e-01 0.243 0.808 |
| DMC 1.497e-02 5.217e-02 0.287 0.775 |
| DC -4.610e-04 7.965e-03 -0.058 0.954 |
| ISI 3.230e-02 2.778e-01 0.116 0.908 |
| --- |
| Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1 |

Reduced Model

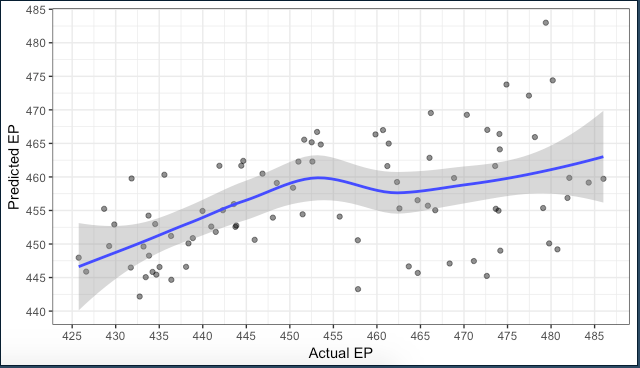
|  |
| --- |
|  |
| Coefficients: |
| Estimate Std. Error t value Pr(>|t|) |
| (Intercept) -969.1284 181.9267 -5.327 3.27e-07 \*\*\* |
| pressure 1.4061 0.1795 7.832 5.81e-13 \*\*\* |
| --- |
| Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1 |

We can see from the tables that the variables with the most significance towards energy production is pressure. Wind is positively related in that an increase can cause energy production to also increase. However, an increase in humidity, will cause energy production to fall, which shows a negative relationship. We can see high p values for both models (above 0.05), which shows that they are not statistically significant for the models. Pressure also had the highest t-value, which is good because it is less likely to have a coefficient be zero by chance.

We also determine the model fit statistics for each model from the training data and measures of prediction accuracy to assess model performance by finding R^2 and RMSE (root mean square error).

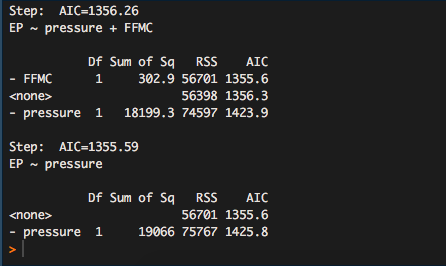
R^2= -0.3405137

RMSE= 1.718122



RMSE is usually negatively valued, thus our value of 1.718122 is higher than normal. The low r^2 merely indicates how poor our model is and how it is not suited to fit our data because it has low variability. The lower the R^2 value, the more disperses the points are from the line. In order to have a good fit, the points ought to be much closer to the line and also have narrow confidence bands.

We need to be able to evaluate metrics to enable us to choose the better model during our selection and one method we employed for that is AIC. AIC stands for Akaike information Criterion and allows us to determine the simplicity of the model and how good of a fit it is in comparison to other models. The results are shown below.



We can see that it produces the same outcome in the reduced model we previously used- that pressure is the most significant variable.

Question 2

Lasso regression which is known as Least Absolute Shrinkage and Selection Operator, is another form of regression that shrinks values (parameter estimates) towards the mean. This produces much simpler models that are also easier to interpret. We first load the package(glmnet) to allow us to fit a linear model with lasso. We then change the variables within the data frame (x variables) into a matrix and perform same for y variables but rather change into a vector for both the test and train data sets.

Train

x\_train <- model.matrix(EP~.,data3\_train )[,-1]

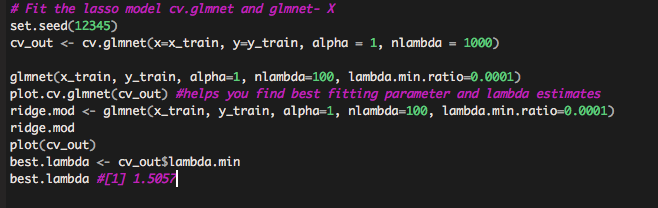
y\_train <- data3\_train$EP

Test

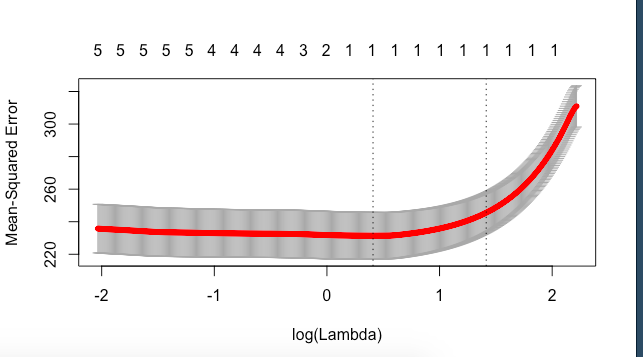
x\_test <- model.matrix(EP~. data3\_test)

y\_test <- data3\_test$EP

We also set the seed once again to (12345) to enable us obtain the same results from the model and input the glmnet function to create a lasso model and cv.glmnet to find the best minimum lambda.



LASSO MODEL



The best lamda derived is 1.5057 since this is a high lambda, the model can be regarded as simple. We then assign the model and the minimum lambda to the test data using a predict function and assess our prediction error, with the output shown below.

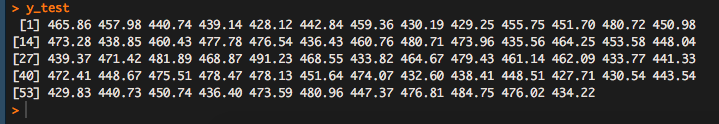
x\_test <- model.matrix(EP~., test)[,-1]

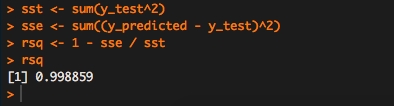
y\_test <- test$EP

y\_predicted <- predict(ridge.mod, s = best.lambda, newx = x\_test)

y\_predicted

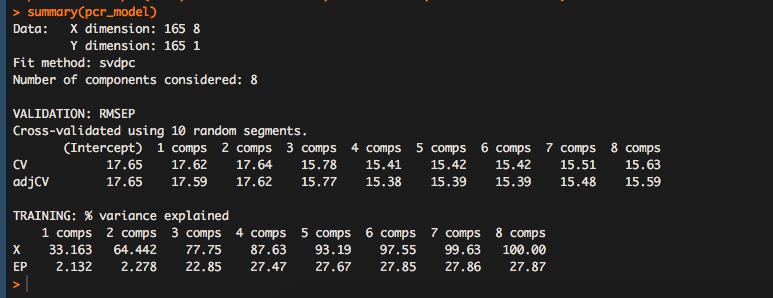
y\_test



To further assess our model, we analyze the prediction accuracy by finding the sum of total squares and the error, to give a total r^2 value of 0.9988589. We have a high R^ value, this model seems to be a better fit as compared to the other models. 

Question 3

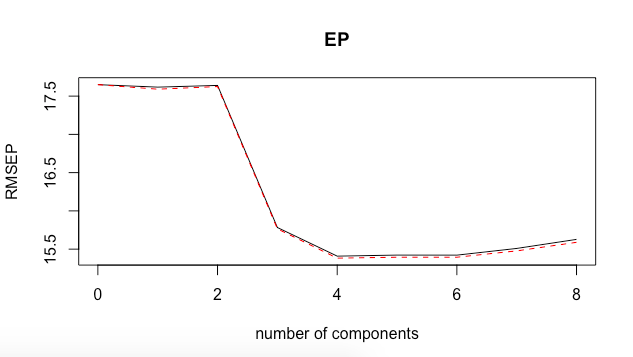
Now we will try to the dimension reduction approach using the PCR model (Principal Components Regression), to get the true meaning of the data. We calculate the principal components and use some as predictors in a linear regression model, but we first make sure these predictors are standardized. Only a few components are needed to describe most of the variability in the data. We first fit a PCR model into our training data set and set the seed to (1000) to maintain consistency in results for the model. We derive a summary which is shown below.



We then plot the RMSE, MSE and R^2 to see how many principle components should be used in our final model. The results are shown below

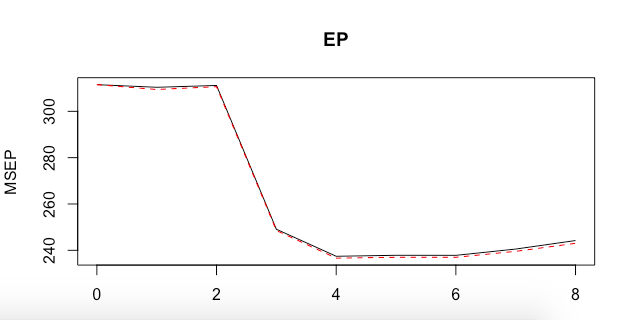
Plot the root mean squared error

validationplot(pcr\_model, val.type = c("RMSEP"))



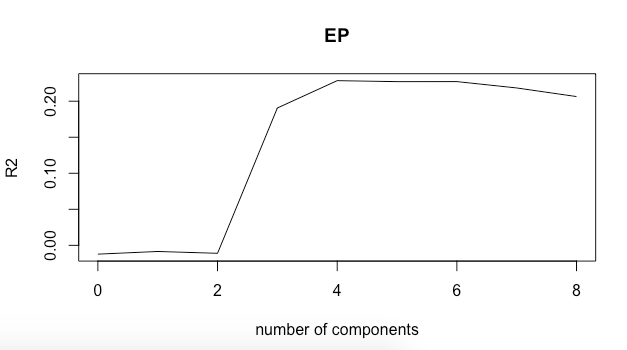
Plot the mean squared error:

validationplot(pcr\_model, val.type = c("MSEP"))



Plot the R-squared:

validationplot(pcr\_model, val.type = c("R2"))



3 principal components were chosen so we test how well they perform with our train and test data sets.

pcr\_model <- pcr(EP~., data = data3\_train,scale =TRUE, validation = "CV")

pcr\_pred <- predict(pcr\_model, test, ncomp = 3)

mean((pcr\_pred - test$EP)^2)

We obtain a mean of 269.5984 an R^2 of 0.99 and an RMSE of 1.71.

The PCR model performed similarly to the other models even with just a few components used.

References

* How and when: Ridge regression with glmnet. (2017, April 10). Retrieved from https://www.r-bloggers.com/how-and-when-ridge-regression-with-glmnet/
* Introduction: Classification, Learning, Features, and Applications. (2011). Wiley Series in Probability and Statistics An Elementary Introduction to Statistical Learning Theory, 1-9. doi:10.1002/9781118023471.ch1
* Rodgers. J (2018). *Intermediate Analytics,* [Powerpoint slides]. Retrieved from https://northeastern.blackboard.com/bbcswebdav/pid-18275609-dt-content-rid-41862042\_1/courses/ALY6060.80796.201835/Week%203.mp4