Dimensionless Technologies Private Limited
Visit us at: www.dimensionless.in
- info@dimensionless.in
- 9923170071, 8108094992

DIMENSIONLESS TECHNOLOGY

Assignment – CART

STATE DATA REVISITED

We will be revisiting the "state" dataset. This dataset has, for each of the fifty U.S. states, the population, per capita income, illiteracy rate, murder rate, high school graduation rate, average number of frost days, area, latitude and longitude, division the state belongs to, region the state belongs to, and two-letter abbreviation. This dataset comes from the U.S. Department of Commerce, Bureau of the Census.

Load the dataset into R and convert it to a data frame by running the following two commands in R:

data(state)

statedata = data.frame(state.x77)

After you have loaded the data into R, inspect the data set using the command: str(statedata)

This dataset has 50 observations (one for each US state) and the following 8 variables:

- **Population** - the population estimate of the state in 1975
- **Income** - per capita income in 1974
- **Illiteracy** - illiteracy rates in 1970, as a percent of the population
- **Life.Exp** - the life expectancy in years of residents of the state in 1970
- **Murder** - the murder and non-negligent manslaughter rate per 100,000 population in 1976
- **HS.Grad** - percent of high-school graduates in 1970
- **Frost** - the mean number of days with minimum temperature below freezing from 1931–1960 in the capital or a large city of the state
- **Area** - the land area (in square miles) of the state

We will try to build a model for life expectancy using regression trees, and employ cross-validation to improve our tree's performance.

Problem 1.1 - Linear Regression Models

Create a **linear regression** model.

First, predict *Life.Exp* using all of the other variables as the independent variables (*Population, Income, Illiteracy, Murder, HS.Grad, Frost, Area)*. Use the entire dataset to build the model.

What is the **adjusted** R-squared of the model?

Problem 1.2 - Linear Regression Models

Calculate the sum of squared errors (SSE) between the predicted life expectancies using this model and the actual life expectancies:

Problem 1.3 - Linear Regression Models

Build a second **linear regression** model using just *Population, Murder, Frost, and HS.Grad* as independent variables (the best 4 variable model from the previous homework). What is the **adjusted** R-squared for this model?

Problem 1.4 - Linear Regression Models

Calculate the sum of squared errors again, using this reduced model:

Problem 1.5 - Linear Regression Models

Which of the following is correct?

- Trying different combinations of variables in linear regression is like trying different numbers of splits in a tree - this controls the complexity of the model.
- Using many variables in a linear regression is **always** better than using just a few.
- The variables we removed were uncorrelated with *Life.Exp*

Problem 2.1 - CART Models

Let's now build a **CART model** to predict *Life.Exp* using all of the other variables as independent variables (*Population, Income, Illiteracy, Murder, HS.Grad, Frost, Area*). We'll use the default *minbucket* parameter, so don't add the *minbucket* argument. Remember that in this problem we are not as interested in *predicting* life expectancies for new observations as we are understanding how they relate to the other variables we have, so we'll use all of the data to build our model. You shouldn't use the method="class" argument since this is a regression tree.

Plot the tree. Which of these variables appear in the tree? Select all that apply.

- [ ] Population
- [ ] Murder
- [ ] Frost
- [ ] HS.Grad
- [ ] Area

Problem 2.2 - CART Models

Use the regression tree you just built to predict life expectancies (using the predict function), and calculate the sum-of-squared-errors (SSE) like you did for linear regression. What is the SSE?

Problem 2.3 - CART Models

The error is higher than for the linear regression models. One reason might be that we haven't made the tree big enough. Set the *minbucket* parameter to 5, and recreate the tree.

Which variables appear in this new tree? Select all that apply.

- [ ] Population
- [ ] Murder
- [ ] Frost
- [ ] HS.Grad
- [ ] Area

Problem 2.4 - CART Models

Do you think the default minbucket parameter is smaller or larger than 5 based on the tree that was built?

- ( ) Smaller
- ( ) Larger

Problem 2.5 - CART Models

What is the SSE of this tree?

(This is much closer to the linear regression model's error. By changing the parameters we have improved the fit of our model.)

Problem 2.6 - CART Models

Can we do even better? Create a tree that predicts *Life.Exp* using **only** *Area* , with the *minbucket* parameter to 1. What is the SSE of this newest tree?

Problem 2.7 - CART Models

This is the lowest error we have seen so far. What would be the best interpretation of this result?

- Trees are much better than linear regression for this problem because they can capture nonlinearities that linear regression misses.
- We can build almost perfect models given the right parameters, even if they violate our intuition of what a good model should be.
- Area is obviously a very meaningful predictor of life expectancy, given we were able to get such low error using just Area as our independent variable.

Problem 3.1 - Cross-validation

Adjusting the variables included in a linear regression model is a form of model tuning. In Problem 1 we showed that by removing variables in our linear regression model (tuning the model), we were able to maintain the fit of the model while using a simpler model. A rule of thumb is that simpler models are more interpretable and generalizeable. We will now tune our regression tree to see if we can improve the fit of our tree while keeping it as simple as possible.

Load the *caret* library, and set the seed to 111. Set up the controls exactly like we did in the lecture (10-fold cross-validation) with *cp* varying over the range 0.01 to 0.50 in increments of 0.01. Use the *train* function

to determine the best *cp* value for a CART model using all of the available independent variables, and the entire dataset statedata. What value of cp does the train function recommend? (Remember that the train function tells you to pick the largest value of cp with the lowest error when there are ties, and explains this at the bottom of the output.)

Problem 3.2 - Cross-Validation

Create a tree with the value of *cp* you found in the previous problem, all of the available independent variables, and the entire dataset "statedata" as the training data. Then plot the tree. You'll notice that this is actually quite similar to the first tree we created with the initial model. Interpret the tree: we predict the life expectancy to be 70 if the murder rate is greater than or equal to ---------- and is less than ----------

Problem 3.3 - Cross-Validation

Calculate the SSE of this tree:

Problem 3.4 - Cross-Validation

Recall the first tree (default parameters), second tree (minbucket = 5), and the third tree (selected with cross validation) we made. Given what you have learned about cross-validation, which of the three models would you expect to be better if we did use it for prediction on a test set? For this question, suppose we had actually set aside a few observations (states) in a test set, and we want to make predictions on those states.

- The first model
- The second model
- The model we just made with the "best" cp

Problem 3.5 - Cross-Validation

At the end of Problem 2 we made a very complex tree using just Area. Use *train* with the same parameters as before but just using Area as an independent variable to find the best cp value (set the seed to 111 first). Then build a new tree using just Area and this value of cp.

How many splits does the tree have?

Problem 3.6 - Cross-Validation

The lower left leaf (or bucket) corresponds to the lowest predicted Life.Exp of 70. Observations in this leaf correspond to states with area greater than or equal to ---------- and area less than ----------

Problem 3.7 - Cross-Validation

We have simplified the previous "Area tree" considerably by using cross-validation. Calculate the SSE of the cross-validated "Area tree", and select all of the following correct statements that apply:

- [ ] The best model in this whole question is the first "Area tree" because it had the lowest SSE.
- [ ] The Area variable is not as predictive as Murder rate.
- [ ] Cross-validation is intended to decrease the SSE for a model on the training data, compared to a tree that isn't cross-validated.
- [ ] Cross-validation will always improve the SSE of a model on unseen data, compared to a tree that isn't cross-validated.