

Lab 4: Bike Sharing Demand

Isabel Casas

Caret Package

caret is a very comprehensive package that wraps many models programmed in R and expands their predicting performance by allowing different data split strategies, hyperparameters tuning and model assessment measures.

Click on this [Link](#) and answer:

1. How many different models do *caret* wrap?
2. Out of all those options, how many different random forest models does *caret* wrap?
3. How many tuning parameters (hyperparameters) do each of these random forest models have?
4. Out of all those options, how would you train the **Oblique Random Forest** using *caret*?

Bike Sharing Demand

Bike sharing is becoming increasingly popular in cities all around the world, providing people with an affordable and flexible way to get around. From a business point of view, being able to predict how many bikes people will need and where they will need them is really important. This helps bike sharing companies make sure that there are enough bikes in the right places, which makes it easier for people to use them. By using a technique called Random Forest regression, we can look at data from bike sharing services and figure out what factors influence how many bikes people need. This can help bike sharing companies and city planners make better decisions about where to put bike lanes and bike parking, which can make it easier and safer for people to ride bikes. This is good for the environment, because bikes don't produce pollution like cars, and it's good for people's health, because biking is a great way to get exercise.

So, by analyzing bike sharing data with machine learning techniques, we can help make bike sharing systems work better for everyone, which is good for the planet and good for people.

Bike Sharing Demand is a dataset that contains hourly and daily count of rental bikes between years 2011 and 2012 in **Capital bikeshare** system with the corresponding weather and seasonal information. The dataset has 17379 observations and 16 input variables (predictors) such as temperature, humidity, wind speed, weather condition, and so on, and 1 output variable (target) which is the number of bikes rented. Learn from the dataset description at the website what each variable represents.

The objective is to find the best tuned random forest using the *caret* package.

Exercise 1

1. Download dataset from the website and upload file *day.csv* to Posit Cloud. Write below the list of variables and their description.
2. Open *day.csv* in R and put its content into variable *day*. Write your code in the chunk below.
3. How many rows and columns does *day.csv* have? Write the code in the chunk below.
4. Discover the variable names of your database. Write the code in the chunk below.
5. Select the relevant variables into a new dataset called *bike*. These variables are: "season", "yr", "mnth", "holiday", "weekday", "workingday", "weathersit", "temp", "atemp", "hum", "windspeed", "cnt" (response).

#2

#3

#4

#5

Activity 1

Let us use the *holdout strategy* to split the dataset into training and testing sets. Then tune the random forest model to obtain the optimal number of *mtry* to fit our data set

Exercise 2

1. Split the dataset into training (70%) and testing (30%) by using the holdout methodology. After this, you should have two datasets, one to train the model and another for prediction. Write code in the chunk below.
2. Using function *train*, train a random forest tuning the *mtry* from a grid of numbers. Save the model into *model1*. Write code in the chunk below.
3. What is the best *mtry* for this problem? Write code in the chunk below.
4. Evaluate the MSE, RMSE and MAE of your training model. Write you code in the chunk below.

#1

#2

#3

#4

Exercise 3

1. Using *model1* predict values in the testing set. Write code in the chunk below.
2. Evaluate the MSE, RMSE and MAE for this prediction. Write code in the chunk below.

#1

#2

Activity 2

Now, we are going to use cross-validation to find the best model while tuning the random forest *mtry*.

Exercise 4

1. Using the training sample from above and the *train* function, fit a random forest but now using a *k*-fold cross-validation. To do this, add *trControl* = *trainControl(method = "cv", number = 10)* to the *train* function. Call *model2* the resulting model. Write code in the chunk below.

2. Prediction with *model2* of your testing data. Write code in the chunk below.
3. Evaluate the MSE, RMSE and MAE for this prediction and compare with the prediction of *model1*. Write code in the chunk below.

#1

#2

#3

Activity 3

Learning to estimate and predict with the multilayer perceptron (MLP).

Exercise 5

Using the training data, train the following models and answer the questions: 1. Multilayer perceptron with one layer (call this *mlp1*). Which is the best number of nodes in the layer for this problem? 2. Multilayer perceptron with one layer (call this *mlp2*). Which is the best learning rate and number of nodes/layer for this problem? 3. Multilayer perceptron with 3 layers (call this *mlp3*). which is the best number of nodes in the layer?

Exercise 6

Third, using Lecture 5 notes, do the following:

1. Plot each of the three MLPs.
2. Predict values using data in your testing sample.
3. Compare the three models prediction performance, which is best?
4. Compare the prediction of the best MLP model with *model1* and *model2*.