1 newpage

2 Data Science: Bridging Principle and Practice

2.1 Part 7: Linear Regression Model (Bike Sharing case study)

2.2 7. The Regression Model

2.2.1 7a. Explanatory and response variables

EXERCISE: Since we want to try the model on the test data as well, we will also perform the same transformations on the test set so it can fit the model. Fill in the code to first select the explanatory variables "temp", "is work day", and "season", then convert the explanatory table to the matrix format using format_X.

Hint: we'll need to go through the exact same steps as in the above cell for the training data, but any references to training data should be replaced by their test data counterparts.

```
In [5]: # select the explanatory variables to use in a DataFrame
        expl_vars = ["temp", "is work day", "season"]

        # convert the explanatory table to the correct format
        X_test = make_X(bike_test, expl_vars)
```

2.2.2 7b. Finding β

EXERCISE: We also want to make predictions for the test data using the β we found during training. Replace the ... in the cell below with an expression to calculate the predictions for the test set. Remember- you need to use @ to multiply each row of explanatory variables in our test set by the β vector. Look at how predict_train was calculated for a guide.

```
In [9]: # make predictions for the test data using beta and the dot product
        predict_test = X_test @ beta

        # create a new column in our test data with predicted total riders
        bike_test["predicted total riders"] = predict_test
```

2.2.3 7c. Evaluating the model

EXERCISE: Calculate the root mean squared error for the test set. Follow each of the above steps, and look at how it was calculated for the training set for some hints.

Before you run the next cell: would you expect the RMSE for the test set would be higher, lower, or about the same as the RMSE for the training set? Why?

```
In [16]: # calculate the error for each data point
         test_errors = y_test - predict_test

         # square the errors
         test_sq_error = test_errors ** 2

         # take the mean of the squared errors
         test_mean_sq_error = np.average(test_sq_error)

         # take the square root of the mean squared errors
         test_rmse = np.sqrt(test_mean_sq_error)
         test_rmse

Out[16]: 1401.9698142555396
```

2.2.4 Visualizing Error

QUESTION: Based on the plots and root mean squared error above, how well do you think our model is doing? What does the shape of the scatter plot of errors tell us about the appropriateness of the linear model here?

ANSWER: This is a pretty mediocre model. Our RMSE for test and training data is over 1400- that is, when we make a prediction of the number of riders, we are off by an average of ~1400 riders. This seems like a high amount of riders when we consider that the average number of total riders is 4504, and the maximum is 8714. We can also see how off the model is from the scatter plots: the data points are clustered pretty loosely around the regression line, and there's a fair amount of vertical space (the error) between the regression line and many points.

From the error scatter plots, we can see that there might be a non-linear correlation between the explanatory variables we chose and the response variable. Ideally, our residual (error) plot would show the residuals pretty evenly and symmetrically distributed above and below the horizontal line at 0. Instead, we can see that most points near zero are under the horizonal line, and most points near the maximum number of riders are over the line. This indicates that a linear model might not be the most appropriate to predict the number of total riders using our explanatory variables.