Homework 2

Due: end of day, Saturday, January 28

Submission instructions: Submit one write-up per group on gradescope.com.

IMPORTANT: Write names of everyone that worked on the assignment on the submission.

In this assignment you will build a predictive model and see how well your model does against others. As a part of the assignment you will submit predictions of your model to

`https://www.kaggle.com/competitions/busn41204-winter23-homework-2`

See below for details.

You can download files for this assignment from
https://www.kaggle.com/competitions/busn41204-winter23-homework-2

Filename	Description
`Bike_train.csv`	contains data that you will use to build your model
`Bike_test.csv`	contains data for which you will make predictions using your model
`sampleSubmission.csv`	example submission in the correct format

1 Description

In a bike sharing system the process of obtaining membership, rental, and bike return is automated via a network of kiosk locations throughout a city. In this problem, you will try to combine historical usage patterns with weather data to forecast bike rental demand in the Capital Bikeshare program in Washington, D.C.

You are provided hourly rental data collected from the Capital Bikeshare system spanning two years. The file `Bike_train.csv`, as the training set, contains data for the first 19 days of each month, while `Bike_test.csv`, as the test set, contains data from the 20th to the end of the month. The dataset includes the following information:

`daylabel`	day number ranging from 1 to 731
`year`, `month`, `day`, `hour`	hourly date
`season`	1 = winter, 2 = spring, 3 = summer, 4 = fall
`holiday`	whether the day is considered a holiday
`workingday`	whether the day is neither a weekend nor a holiday
`weather`	1 = clear, few clouds, partly cloudy 2 = mist + cloudy, mist + broken clouds, mist + few clouds, mist 3 = light snow, light rain + thunderstorm + scattered clouds, light rain 4 = heavy rain + ice pallets + thunderstorm + mist, snow + fog
`temp`	temperature in Celsius
`atemp`	"feels like" temperature in Celsius
`humidity`	relative humidity
`windspeed`	wind speed
`count`	number of total rentals

Load the data using:

```
bike.test = read.csv('Bike_test.csv')
bike.train = read.csv('Bike_train.csv')
```

2 Questions

1. Before you build your predictive model, let us first explore the data.

   a. Visualize the relationship between `count` and each one of the following variables on a separate scatter plot: `windspeed`, `humidity`, `temp`, and `atemp`.

   b. How does `count` depend on the season? Consider visualizing this relationship with a boxplot.

   c. How does `count` depend on the time of the day (`hour`)? Does this relationship change depending on whether it is a `workingday` or not? A scatterplot could be used to visualize the relationship. You might consider coloring the observations on the scatterplot using the temperature (`temp` or `atemp`) do discern how the temperature affects hourly number of rentals.

   d. Does the relationship between `count` and `hour` change by `season`?

   e. Does the distribution of hourly number of rentals change between 2011 and 2012? What does this tell you about the rental business?

2. Build a model to predict the bikeshare counts for the hours recorded in the test dataset. Save your predictions to a `.csv` file that you will submit to Kaggle (see Kaggle instruction below.) Provide a write-up that explains how you went about building your model. Attach the code to create the submission `.csv` file as an appendix to your homework submission.

   A sample script for generating a valid `.csv` file for submission is given below. This code builds a simple linear model for predicting `log(count)` using all the available variables. You will easily do better than this.

```
bike.train$logcount = log(bike.train$count + 1)
bike.train = subset(bike.train, select = -count )

lm.fit = lm(logcount ~ . , bike.train)
yhat = round( exp(predict(lm.fit, bike.test)) - 1 )

#sampleSubmission is a dataframe with columns 'Id' and 'count'
sampleSubmission = data.frame(Id=1:length(yhat), count=yhat)
write.csv(sampleSubmission,
          file = "sampleSubmission.csv",
          row.names = FALSE,
          quote = FALSE)
```

   Predictions will be evaluated using the root mean squared error (RMSE), calculated as

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(m_i - \hat{m}_i)^2}$$

   where m_i is the true count, $\hat{m}_i$ is the estimate, and n is the number of entries to be evaluated.

   Hints:

   - We have barely scratched the surface with data exploration. It will be helpful to further examine the data graphically to spot any seasonal pattern or temporal trend.

- Are there any unusual patterns in the training or test data?
- There is one day in the training data with weird `atemp` record and another day with abnormal `humidity`. Find those rows and think about what you want to do with them. Is there anything unusual in the test data?
- Think about how you would include each predictor into the model, as continuous or as categorical?
- Is there any transformation of the predictors or interactions between them that you think might be helpful?

3 Kaggle Instructions

Submission files should contain two columns with headers: `Id` and `count`. There should be 6,493 entries of predicted values in total. See code above that generates a valid submission file `sampleSubmission.csv` for reference.

Everyone needs to sign up for the competition before forming the team. Once you have signed up, you can find your teammates and form the team on the Team section of the page.

You can submit up to 10 times everyday and check your relative performance on the public leaderboard.

However, **the final ranking** will be based on the private leaderboard, which will not be released until the end of the competition.

4 Evaluation

You will receive points based on:

- your write-up;
- whether we can create your submission based on the code you provided; and
- your relative ranking in the class.

5 Submission

You need to submit two things:

- a write-up to gradescope
  - includes the name of your team
  - includes code to generate your submission file as an appendix;
- a valid submission to Kaggle.