Project 2

Project Objectives

This project involves creating predictive models and automating Markdown documents. You will then create a blog post linking to your analyses.

Project Work

All project work should be done in a github repo. Ideally, you have connected RStudio with github and can work from the command line within RStudio. All major updates should be made through github so we can track your activity.

Data

You'll read in and analyze an online news popularity data set (previous text is a link). You can read more about the data at the website.

Goal

The goal is to create models for predicting the `shares` variable from the dataset. You will create two models: a linear regression model and a non-linear model (each of your choice). You will use the parameter functionality of markdown to automatically generate an analysis report for each `weekday_is_*` variable (so you'll end up with seven total outputted documents).

Report

At first, consider just using the 'Monday' data. Once you have all of the below steps done for that data, then automate it to work with any chosen day of the week data.

Introduction section

You should have an introduction section that describes the data, the purpose of your analysis, and the methods you'll use (roughly - more detail can be given later in the document).

Data

You should briefly describe the data and the variables you have to work with (no need to discuss all of them, just the ones you want to use).

You should randomly sample from (say using `sample()`) the (Monday) data in order to form a training (70% of the data) and test set (30% of the data). You should set the seed to make your work reproducible.

Summarizations

You should produce some basic (but meaningful) summary statistics about the training data you are working with. The general things that the plots describe should be explained but, since we are going to automate things, there is no need to try and explain particular trends in the plots you see (unless you want to try and automate that too!).

Modeling

Once you have your training data set, we are ready to fit some models.

You should fit two types of models to predict the `shares`. One model should be an ensemble model (bagged trees, random forests, or boosted trees) and one should be a linear regression model (or collection of them that you'll choose from).

The article referenced in the UCI website mentions that they made the problem into a binary classification problem by dividing the shares into two groups (< 1400 and ≥ 1400), you can do this if you'd like or simply try to predict the shares themselves.

Feel free to use code similar to the notes or use the `caret` package.

After training/tuning your two types of models (linear and non-linear) using cross-validation, AIC, or your preferred method (all on the training data set only!) you should then compare them on the test set. Your methodology for choosing your model during the training phase should be explained.

Automation

Once you've completed the above for Monday, adapt the code so that you can use a parameter in your build process that will cycle through the `weekday_is_*` variables.

Blog Post and Repo Stuff

On your project repo you should go into the settings and enable github pages (feel free to select a theme too!). This will make it so your repo can be accessed like your blog (username.github.io/repo-name).

Knit your .Rmd file with the output type `output: rmarkdown::github_document` in the YAML header. This will create a .md file which will automatically be rendered by github (you'll have seven of these).

In the README.md file for the repo, create links to each subdocument with a brief description. Links can be made using relative paths. For instance, if you have all of the outputted .md files in the main directory you would just use markdown linking:

The analysis for `[Monday is available here](MondayAnalysis.md)`.

Of course, this supports the use of folders as well if you output the files into separate folders.

Once you've completed the above tasks you should write a brief blog post outlining your project and linking to the username.github.io/repo-name site. You should then also reflect on the process you went through for this project. Discuss things like:

- what would you do differently?
- what was the most difficult part for you?
- what are your big take-aways from this project?

Rubric for Grading

Item	Points	Notes
Introduction	10	Worth either 0, 3, 7, or 10
Data split	5	Worth either 0, 2, or 5
Ensemble model fit	15	Worth either 0, 5, 10, or 15
Linear regression fit	15	Worth either 0, 5, 10 or 15
Model selection	15	Worth either 0, 3, 6, 9, 12, or 15
Test set prediction	5	Worth either 0, 2, or 5
Automation	10	Worth either 0, 3, 6, or 10
Conclusions	15	Worth either 0, 5, 10, or 15
Blog post and repo setup	10	Worth either 0, 4, 6, or 10

Notes on grading:

- For each item in the rubric, your grade will be lowered one level for each each error (syntax, logical, or other) in the code and for each required item that is missing or lacking a description.

- Although not explicity in the criteria above, points will be taken off for not following good programming practices (up to 30), for not using appropriate markdown options/formatting (up to 20), and for not using github as you work through the project (up to 30).