Project 2

Project Objectives

This is a group project (see the project 2 page) that involves creating predictive models and automating Markdown reports. Once you've completed the project you will also create a blog post linking to your analyses.

Project Work

The first step is for the *first* group member to create a github repo and add the *second* group member as a collaborator. The second group member then needs to accept the membership. This gives everyone access to push changes up to the repository. **All project work should be done within this repo.**

Each time you go to work on the project, you should pull down any of the latest changes using `git pull`. You should then upload any changes you've made via the usual workflow done previously. There may occasionally be merge conflicts that have to be dealt with. This can be done with the `Git` tab in RStudio. Let us know if you are having issues with conflicts that you can't resolve!

Repo Setting

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). Be sure to choose the master or main branch as the one to use if you have choices there.

You'll be automating the creation of documents using R Markdown (one for each `data_channel_is_*` setting, i.e. type of article). Each document should be rendered as a `github_document` from a single .Rmd file. In the README.md file you should create links to each of the documents you will create (Lifestyle analysis, Entertainment analysis, etc.). Links can be made to the sub-documents 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 `[Lifestyle articles is available here](LifestyleAnalysis.html)`. Note we link to the `html` file even though the file we create is a `.md` file - github creates the .html for us.

In the repo's README.md file (which doesn't need to be created from a .Rmd file, just use the one you initialize into the repo) give a brief description of the purpose of the repo, a list of R packages used, links to the generated analyses, and the code used to create the analyses from a single .Rmd file (`render()` code).

Blog

Once you've completed the projct each of you should write a brief blog post outlining your project and two links to the username.github.io/repo-name site and the repo itself (the username may correspond to your partner). You should then also reflect on the process you went through. Discuss the following:

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

Topic

What are you actually doing? You'll read in and analyze an online news popularity data set. You'll subset the data by `data_channel_is_*` (one of six groups). Then you'll summarize the data and try to predict the number of `shares` using predictive models.

Report

Recommendation: At first, consider just using data from a single `data_channel_is_*` source. Once you have all of the below steps done for that data, then you can automate it to work with any chosen data

channel. Note: It may be easier to create a single variable representing the data channel when automating the subsetting (although there are many ways to do this).

- All code chunks should be shown unless they are setup code chunks.

Introduction section

You should have an introduction section that briefly describes the data and the variables you have to work with (just discuss the ones you want to use). Your target variables is the `shares` variable.

You should also **mention** the purpose of your analysis and the methods you'll use to model the response. You'll describe those in more detail later.

This section should be done by the 'second' group member.

Data

Use a relative path to import the data. Subset the data to work on the data channel of interest.

This section should be done by whoever can get to it first.

Summarizations

You should produce some basic (but meaningful) summary statistics and plots about the training data you are working with (especially as it relates to your response).

As you will automate this same analysis across other data, you can't describe the trends you see in the graph (unless you want to try to automate that!). You should describe what to look for in the summary statistics/plots to help the reader understand the summary or graph. Ex: A scatter plot with the number of shares on the y-axis and the positive word rate on the x-axis is created:

```
'We can inspect the trend of shares as a function of the positive word rate.  If the points
show an upward trend, then articles with more positive words tend to be shared more often.
If we see a negative trend then articles with more positive words tend to be shared less often.'
```

Each group member is responsible for producing some summary statistics (means, sds, contingency tables, etc.) and for producing at least three graphs of the data.

Modeling

You'll need to split the data into a training (70% of the data) and test set (30% of the data). Use `set.seed()` to make things reproducible.

The goal is to create models for predicting the number of shares in some way. Each group member should contribute a linear regression model and an ensemble tree-based model. As we are automating things, describing the chosen model is tough, so no need to worry about that.

The first group member should fit a random forest model and the second group member should fit a boosted tree model. Both models should be chosen using cross-validation.

Prior to the models fit using linear regression, the first group member should provide a short but thorough explanation of the idea of a linear regression model.

Prior to each ensemble model, you should provide a short but reasonably thorough explanation of the ensemble model you are using (so one for each group member).

Comparison

All four of the models should be compared on the test set and a winner declared (this should be automated to be correct across all the created documents).

This can be done by one group member and the automation done by the other (see below).

Automation

Once you've completed the above for a particular data channel, adapt the code so that you can use a parameter in your build process. You should be able to automatically generate an analysis report for each `data_channel_is_*` variable - although again, you may want to create a new variable to help with the subsetting. You'll end up with six total outputted documents.

This should be done by the group member that doesn't automate the comparison of models part.

Submission

In the project submission, you should simply put a link to your blog post (which will have a link to your github pages and github repo).

Group Issues

Please notify me ASAP of any group member issues. You should look over your partner's work/explanations and discuss that with them if you have any issues with what they've done. Both group members are graded on all the work done regardless of who was assigned to do it.

Rubric for Grading (total = 100 points)

Item	Points	Notes
Introduction	10	Worth either 0, 5, or 10
Data split	5	Worth either 0 or 5
Summarizations & discussions	20	Worth either 0, 5,. . . , or 20
Modeling, selection, & discussion	35	Worth either 0, 5, . . . , 35
Test set prediction and automation	10	Worth either 0, 5, or 10
Automation	15	Worth either 0, 5, 10, or 15
Blog post and repo setup	10	Worth either 0, 5, 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.
- **If your work was not completed and documented using your github repo you will lose 50 points on the project.**
- You should use Good Programming Practices when coding (see wolfware). If you do not follow GPP you can lose up to 25 points on the project.
- You should use appropriate markdown options/formatting (you can lose up to 20 points) for not doing so.