**Regression Project**

* **Get a dataset from Kaggle (not too hard, not too easy). It must contain at least one categorical explanatory variable and the variable that you are trying to predict must be numerical.**
* **You will analyze it using scikit-learn and then post about your analysis on your GitHub blog. Your blogpost should link to your GitHub regression code repo.**

**Some things to discuss**:

1. Did your data need cleaning? If so, what was your process? Justify any decisions.
2. Use a one-hot matrix to transform your categorical variable and add it to your input.
3. Use .corr and .sort\_values to sort your variables according to correlations with the response (y) variable. What explanatory variables were most positively correlated with your response variable? What variables were most negatively correlated?
4. Is your response variable skewed? If so, how did you transform it? Did that help your predictions?
5. Use sns.pairplot to plot scatterplots of your variables versus each other. Choose an explanatory variable that does not appear to be linearly related to your response variable. Try to come up with an optimal degree polynomial that fits the data better. Include a plot of your degree versus test error and clearly show the U-shape. If there is a polynomial that fits the data better than the linear model, add the polynomial features of this variable to your input matrix.
6. Use a pipeline to scale your data before applying RidgeCV.
7. What alpha was best?
8. What is the equation of your model? Use variable names instead of y,x1,x2,x3…. If there are many variables, you can list the first three and then do “…”.
9. What is the R^2, adjusted R^2, and MSE on your train and testing data? They should be close, or else you are overfitting.
10. Explain what the words “bias” and “variance” mean in the context of your model.

**Make sure that your code is clean enough for me to read (put markdown cells in between describing work) and make sure that your blog is clear enough for a lay-person to read. Discuss what is interesting or unexpected in the context of your data.**

**GitHub blog instructions:**

1. Open the Hello World post located inside your Machine\_Learning/username.github.io/\_posts directory using a text editor such as TextEdit of Sublime Text or Notepad. You’ll want to copy this and save it as 2019-10-15-Regression.md. This is where you will write your blogpost. If you want to include any pictures, save them in the images directory and then type something like the following where you want to include a photo in your text file:

<img src="/images/picture1.png" width="600"/>

Save this text file.

1. Open Applications – Terminal.
2. You need to move into the github.io directory. You may have put this inside your machine learning directory inside your Documents or Desktop. For example, you might type:

cd Desktop/Machine\_Learning/username.github.io

You can also type changes one at a time and then view what subdirectories are located there. For example: cd Desktop, then ls (for list directory), then cd Machine\_learning, then ls.

1. Type pwd for “print working directory” and make sure that the path ends in /username.github.io.
2. Okay, now you are ready to commit your changes to GitHub. Type one at a time:

git add -A

git commit -m ‘editing blog’

git push -u origin master

1. Wait about 30 seconds and then refresh your username.github.io website and you should see the changes reflected.
2. Each time you want to update your blogpost, you’ll repeat instructions 2-6.

**GitHub code repo instructions (putting your Python code up on the web).**

1. Click on “Start a project” or the “+ sign – New Repository” in the top right hand corner of the page. Either one will bring you to the same page.

Create a Repository named the **exact** same thing as your local regression project folder. For example, “Regression\_Project” and write in the description something like “Regression project for machine learning class.” Keep the project public and **don’t click** on the README box. Then click on “Create repository”.

The webpage that you’ll be brought to will give you instructions for creating a new repo. Keep this webpage open but don’t follow the instructions for now. Instead, do this next…

**2. Get into this directory on your local computer.**

In your terminal, type something like “cd Desktop/Machine\_Learning/Regression\_Project” and press enter

In your terminal, type “pwd” and make sure it ends in Regression\_Project

**3. Create a README file**

It is best practice to always include a README.md file in your repo. This is a file that explains what your code is and how it works. Note that .md stands for “markdown”. Markdown is a way to style text on the web. If you want to read more about it (it’s optional) you can go to:

https://guides.github.com/features/mastering-markdown/

**In your terminal, type:**

1. touch README.md *(then press enter)*

2. vim README.md *(then press enter)*

3. Type “i” to get into the insert mode

4. Write your file description. Something like “This is a regression project for my machine learning class.”

5. Hit escape and then type “:w” enter and then “:q” enter to save and exit vim.

**4. Synchronizing your local** Regression\_Project **folder with the GitHub web-based** Regression\_Project **folder**

Okay, these next five steps are the most important. TYPE THEM IN EXACTLY. If you make even the slightest error it can be very difficult unless you are a Git expert (which I’m not, so even I won’t be able to help you!) **PROOFREAD EACH LINE 5 TIMES BEFORE YOU PRESS ENTER!**

First to check that you are in the right place: **type “pwd” and it should say “Regression\_Project”.** Then from there, type:

1. git init *(then press enter)*

2. git add –A *(then press enter* - note this says add ALL the files in this folder to GitHub)

3. git commit –m “first commit” *(then press enter -* note that you can put whatever you want inside the quotation marks, but it is standard on the first commit to just say “first commit”)

It should say something like “Your name and email address were configured automatically.”

Go to the GitHub website. It should say something like “Quick Setup: If you’ve done this before” with a long url. Copy and paste that url and stick it into the following place inside your terminal:

4. git remote add origin *copied\_url*

Note: if you DO make a mistake entering the url, you can fix it by typing:

git remote set –url origin *the correct url*

5. git push –u origin master *(then press enter)*

At this point, it might ask for the GitHub username and password that you already set up.

**5. Look at your fancy pants webpage**

Return to the GitHub webpage that you have open and click on Regression\_Project at the top of the page. You should now see your regression project files on this webpage. Everyone can see your wonderful work!

**6. Troubleshooting**

If your website didn’t update correctly, you may have accidentally typed “git init” outside of the Regression folder. If so, do this:

1. While you are still inside the wrong folder, type: rm –rf .git

2. Do the directions again in Step 7 again while you are in the CORRECT folder

Other errors? Copy and paste the error code and “github” into google to see what solutions come up.

**Linking your code repo to your blog post:**

**At the end of your blog post, type something like the following (replace username with your username):**

**You can find my code located [here](**[**https://github.com/username/Regression\_Project**](https://github.com/username/Regression_Project)**)**

This will include a hyperlink to your code repo. Any other hyperlinks you want to include can be formatted the same way.