**Using Github With Colab**

1. Make sure you have access to the Medlytics repo (https://github.com/BeaverWorksMedlytics). If you don’t, give your Git username to one of the instructors and we will invite you.
2. Make sure you have Terminal/GitBash open and ready
3. Fork the Week1 repo (<https://github.com/BeaverWorksMedlytics/Week1_public>) to your personal Github. Fork option is in the top right corner of the repo page.
4. Next we will learn how to use Github with Google Colaboratory (aka “Colab”). You are not required to use Colab and can instead use your favorite IDE on your local machine, but we urge you to use Colab due to its free GPU support and ease of collaboration (real-time code updating).
5. Open Google Colab (<http://colab.research.google.com>) and select:

File > Open Notebook > select Github tab > type your github username >

search > select the Week1 repo you have forked >

select the Intro\_Colab\_Github\_Test.ipynb from the Github\_Colab\_Intro folder

This should open an empty notebook in Colab.

1. Make a change to the notebook in Colab (anything!)
2. To commit this change to your personal Github repo, select:

Files > Save a Copy in Github > select your forked repo >

enter a commit message explaining your changes succinctly > Click OK.

This is effectively the same as a normal commit.

To access the changed file on your local machine, simply git pull.

To push changes to a file on your local machine and push it to Github:

git add <filename>

git commit –m “Commit Message”

git push <remote\_name> <branch\_name>

To access it in Colab, reopen the notebook the same way we did in step 5. This is effectively Colab’s version of a pull.

1. Up to now we’ve successfully forked a repo, pushed and pulled changes using Colab! For the most part, this is all you will need to do when working on lesson notebooks. But when working on challenges and your Week 4 project, you might need to do some more complicated code sharing and integrating.
2. During challenges and the Week 4 project, we recommend that one person in each group hosts the notebook in progress in their personal Github using a public repository and the other members of the group clone that repo and push/pull commits.
3. Feel free to use branches instead of always committing to master. Google Colab cannot do anything fancier than commits and pulls so you will have to use Terminal/GitBash for this.
   * Create branch in Terminal/GitBash (git checkout –b <branch name>)
   * Push the branch (git push origin <branch name>)
   * You can now push to this branch (using Terminal/GitBash or Colab) instead of master. Then when you are sure everything is working and ready to commit to master, merge the branch with master from the master directory (git merge <branch name>). You can then delete your branch locally (git branch –d <branch name>) and remotely (git push origin --delete <branch name>).
4. If we need to change something in the original BeaverWorksMedlytics repo during in the middle of a lesson or challenge that you have already made progress on, you will need to sync your forked repo with the original. This can be a little complicated.
   1. First, in Terminal/GitBash, from the the working directory of your project (master) run ‘git fetch upstream’
      * If you get an error, run ‘git remote add upstream <original repo url>’
   2. Then run ‘git fetch upstream’
   3. Next, checkout master (git checkout master) and run ‘git merge upstream/master’
      * This will add and commit the updated version of master
   4. Lastly, ‘git push <origin><master>’ and you can then pull from your fork using Colab.

That’s all! Feel free to come back to this tutorial later on if you forget how to sync your fork or make a branch.

