Homework 5 - Object Classification

When preparing your homework submissions, don't forget to `git fetch, git status, git pull` *before you issue any other commands in Git – this is to guard against you changing a document that someone else is working on in the same directory*[1].

Don't forget to `git add` *and* `git commit` *(with* `-m` *comments) frequently as you work. This allows other users to see how your work progressed and it automatically backs up your work as you produce. Thus, you're less likely to lose any of your work and/or so you can revert to earlier versions of your work as needed.*

Remember to comment your code carefully with your initials before every comment (as in # JCR `I just wrote a Python comment to document a change.` *Remember to provide an informative header for* **every** *function that you write. Also provide a README file to inform people how to run your code.*

Homework

In my `week11` directory in Git is a file `qsos-ra180-dec30-rad3.fits` that contains the coordinates of 316 confirmed quasars. An astronomer is conducting a large spectroscopic survey of the sky. The purpose of obtaining the "test" information in `qsos-ra180-dec30-rad3.fits` was to better determine which objects in a further, larger spectroscopic sky survey should be targeted as quasars. The survey will be limited to a g-band magnitude of 20 ($g < 20$).

Using my file `qsos-ra180-dec30-rad3.fits` determine color cuts and flag cuts that can be used to target quasars. You are allowed to use WISE $W1$ and $W2$ and SDSS $ugriz$, as well as any flag cuts in the SDSS sweep files[2]. You are *n*ot allowed to use anything other than simple color and flag cuts, so do not apply `extreme deconvolution` or any advanced statistical algorithm.

Write code that **applies** the cuts you have found to classify which of a set of passed objects are quasars. Your code should include a function that takes, as a single input, a rec array that has columns `RA` and `DEC`, and the SDSS sweeps columns `PSFFLUX`, `EXTINCTION`, `RESOLVE_STATUS`, `OBJC_TYPE`, `OBJC_FLAGS`, `OBJC_FLAGS2`, `FLAGS`, and `FLAGS2` columns and the WISE `W1_NANOMAGGIES` and `W2_NANOMAGGIES` columns. Your code should return a single array that contains a "1" for those indexes of the input structure that should be targets as quasars and a "0" for those that should not. So, say the input structure were to contain 9 rows. Then your output should be something like:

```
array([0, 0, 1, 0, 0, 1, 0, 0, 0])
```

Or, if the input structure were to contain 13 rows, then your output might be:

```
array([1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0])
```

Make sure that your code includes a function that requires no additional input beyond a rec array with the stated 12 columns and that your code can handle such an input without raising

[1] This shouldn't be a big deal unless we're working collaboratively, but you should get into the habit *now*.

[2] essentially anything in directory `/astro/astr8020/dr15/eboss/sweeps/dr13_final/`

an error. If the input rec array that I pass to your code is called `objs`, then I should be able to call your code at the Python prompt as, e.g.:

```
>> from your_code import awesome_function
>> out_array = awesome_function(objs)
```

Also make sure that your code produces the requested array output. This should be returned as an output from a function and should *not* for example, just print "0s" or "1s" to the screen. **If a function in your code cannot handle the specified rec array as an input and does not produce the specified output, then you may receive zero points for accuracy.**

This week's homework will be assessed objectively, except for commenting which will be assessed in the usual way. Moving forward, I will no longer be assessing the structure or compactness of your code. This week, accuracy will be worth a total of 30 points, and speed and commenting will be worth 10 points each. Accuracy and speed will be assessed according to benchmarks:

- Accuracy: I will apply your algorithm to a region of the sky I know of that has a high density of confirmed quasars, and I will determine your score out of 30 points using the following formula:

  $$q = \text{the total number of confirmed quasars you recover in this region per deg}^2$$

  $$t = \text{the total number of objects you target in this region per deg}^2$$

  $$f = \frac{q}{t}$$

  $$\text{IF } f > 0.85 \text{ THEN SET } f = 0.85 \text{ ... IF } q > 10 \text{ THEN SET } q = 10$$

  $$\text{Score} = 30 \times \frac{f}{0.85} \times \frac{q}{10}$$

  In other words, if you correctly find more than 10 quasars per square degree and more than 85% of your overall targets are confirmed quasars, you receive 30 points. If you find fewer than 10 quasars per square degree and/or fewer than 85% of your overall targets are confirmed quasars, your score will be scaled down. For instance, if 77% of your overall targets turn out to be quasars and you find 8.5 quasars per square degree, your score would be $30 \times (0.77/0.85) \times (8.5/10) = 23.1$ out of 30 points.

- Speed: Speed will be assessed out of a total of 10 points using the following formula:

  $$t = \text{the total number of seconds your code takes to run}$$

  $$\text{IF } t < 2 \text{ then set } t = 2 \text{ ... IF } t > 12 \text{ THEN SET } t = 12$$

  $$\text{Score} = 12 - t$$

  Please include a timer at the very start and the very end of your code that prints out the total time that it takes your code to run.

- Commenting: Will be assessed as usual out of 10 points.