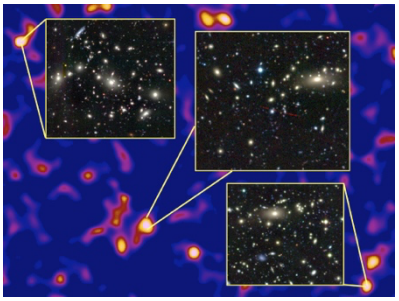
Assignment 1: Swiss Cheese in Space

AI 539, Winter 2023

Due: Jan. 21, 2023, at 11:59 p.m.



Credit: Van Waerbeke, Heymans, and CFHTLens collaboration.

Can you train a machine learning model to distinguish stars from galaxies using telescope observations? This assignment gives you the opportunity to explore and assess different methods for handling missing values in data sets.

Learning objectives:

- You will create a data set profile for data from the Canada–France–Hawaii Telescope Lensing Survey (CFHTLenS) and identify where missing values appear.
- You will compare different methods for handling missing data.
- You will describe your results in a **Findings** document.

Resources and tips:

- Download the data set from Canvas (**`cfhtlens.csv`**), with 5000 sky objects.
- Review the documentation: https://www.cadc-ccda.hia-iha.nrc-cnrc.gc.ca/en/community/CFHTLens/README_catalogs_release_2018-01-31.txt
  - The class label is "star" if "CLASS_STAR" >= 0.5 and "galaxy" otherwise.
  - Only a subset of the features described at this URL are included in cfhtlens.csv.
  - "id" is an object identifier, and "pos" gives its sky position. When training a classifier, use only the 9 features that come after "CLASS_STAR."
  - Missing values are indicated with `99` or `-99` in the magnitude (MAG_*) fields.
  - More information about the scientific value of this data: https://www.cfht.hawaii.edu/en/news/CFHTLens/
- **Good programming style is required (readable code with good use of whitespace and variable names; comment header with author, date, assignment; internal comments to help the reader understand the program). Attribution must given when code is re-used from another source.**

What to do:

1. Write a Python program called **`data_profile.py`** to generate a **data set profile** that includes, for each of the 9 features:
   A. the minimum, maximum, mean, and median values,
   B. a histogram, and
   C. the number of missing values.

2. Investigate the missing values. Are there any **patterns** for which sky objects have missing data? Are they MCAR, MAR, or NMAR? See Emmanuel et al. (2021): https://journalofbigdata.springeropen.com/articles/10.1186/s40537-021-00516-9

3. Write a Python program called **`train_eval.py`** to accomplish steps 3-6:
   Create a training data set by randomly (but reproducibly) selecting 3000 items from those that have **no missing values**. Train a classifier of your choice (e.g., random forest, support vector machine) to predict whether an object is a star or galaxy. Use a library such as scikit-learn to save development time. One way to generate the train/test sets:

```
data = pd.read_csv(csvfile, na_values=…, usecols=…) # Fill in na_values, usecols
Xy = data.to_numpy()
X = Xy[:, 1:]
y = Xy[:, 0] >= … # What value?  See above.
missing = np.sum(np.isnan(X), axis=1) > 0
X_use = X[~missing]
Y_use = y[~missing]
X_train, X_test, y_train, y_test = \
     train_test_split(X_use, y_use, train_size=3000, random_state=0)
```

4. Create a test set with the remaining 2000 items. Note if you use the above method, you'll need to add the items with missing values back in to get all 2000:

```
X_test_full = np.concatenate((X_test, X[missing]))
y_test_full = np.concatenate((y_test, y[missing]))
```

5. Implement the following **five methods** for handling missing values in the test set:
   A. Do not classify items with missing values (i.e., **abstain**). Abstentions count as errors.
   B. Predict the **majority class** (based on the training set) for items with missing values.
   C. **Omit any features** (from training and test) that contain any missing values. Re-train using only the remaining features and evaluate on the test items.
   D. Infer (impute) the missing values using the **average value** for the feature (from the training set). Classify using all features.
   E. Infer (impute) the missing values using **another method of your choice**. Try to achieve higher accuracy than the previous four methods.

   Print out the **classification accuracy** for the (a) **entire test set** and (b) **only the test items that (originally) had missing values.**

6. **Predict the class of your own sky object**. The sky position in "pos" consists of two values, the right ascension (RA) and declination (Dec). Construct an RA that consists of `134.<your_height_in_cm>` and Dec of `-1.97<your_age_in_years>` and search the catalog for an object close to this sky position at https://www.cadc-ccda.hia-iha.nrc-cnrc.gc.ca/en/community/CFHTLens/query.html .

1) Enter your RA and Dec values, specify to search within "10" arc seconds, select "HTML" or "CSV" for "return format", and click the boxes for these features: `id, pos, CLASS_STAR, PSF_e1, PSF_e2, scalelength, model_flux, MAG_u, MAG_g, MAG_r, MAG_i, MAG_z`
2) Modify the SQL query from "`top 10`" to "`top 1`" (to get the closest match) and add "`and fitclass = 0 or fitclass = 1`" to the end of the query (to filter out bad data). Click "Submit query." If you don't get any results, widen the search (increase arc seconds).
3) Classify the resulting feature vector with your classifier. If it has missing values, use the method you identified as best from your study above to handle them.

7. Create a **Findings** document (pdf, doc, docx, rtf, or txt) that includes:
   1. Your data set profile. Did you find any patterns in the missing data? Would you describe them as MCAR, MAR, or NMAR?
   2. Description of the classifier and its parameter values.
   3. Description of the new method (E) that you developed/used.
   4. Test set classification accuracy using each of the missing value methods (A-E). Include two tables (full test set and those with missing values only).
   5. Discussion/comparison of results. Which method do you recommend and why?
   6. Show your sky object's feature vector (plus label). What result did you get when classifying your own sky object? Was it classified correctly or not?
   7. **Reflection**: What did you learn from completing this assignment? How long did this project take? What was the hardest part? What might you use in the future?
   8. Extra credit (*optional*): We already know the class of these sky objects. **What is the utility of training a model to make predictions on this data?**
   9. Extra credit (*optional*): Run the same experiment, but this time choose the 3000 training items from the full data set (i.e., **allow items with missing values**). Report test set accuracy for methods C, D, and E (handle missing values in the training data before training). **Do your conclusions about the best method change?** (Note: accuracies are comparable only when using the same test set.)

What to submit (see rubric on Canvas for point breakdown):

1. `data_profile.py` to generate a data set profile for the CFHTLenS data set.
2. `train_eval.py` to read in the CFHTLenS data set, apply missing data methods, and generate results.
3. A `README.txt` file with your name, date, and assignment; any dependencies (e.g., libraries to be installed); and instructions for how to run each program. You do not need to submit the `cfhtlens.csv` data file.
4. Your Findings document (described above).

Questions? Ask here: https://discord.com/channels/1061031671010447380/1061031671710892039

More information about the CFHTLenS catalog data set:

- Hildebrandt et al. (MNRAS, 2012): https://academic.oup.com/mnras/article/421/3/2355/1078193
- Erben et al. (MNRAS, 2013): https://academic.oup.com/mnras/article/433/3/2545/1236190

Credit statement: This assignment uses observations obtained with MegaPrime/MegaCam, a joint project of CFHT and CEA/DAPNIA, at the Canada-France-Hawaii Telescope (CFHT) which is operated by the National Research Council (NRC) of Canada, the Institut National des Sciences de l'Univers of the Centre National de la Recherche Scientifique (CNRS) of France, and the University of Hawaii. This research used the facilities of the Canadian Astronomy Data Centre operated by the National Research Council of Canada with the support of the Canadian Space Agency. CFHTLenS data processing was made possible thanks to significant computing support from the NSERC Research Tools and Instruments grant program.