MLD 2024 - Evaluation problem set

Deadline: February 25

This problem set is the one that will be graded for this course. It consists of three problems (I'll refer to them as questions in the following). You need to choose **one** of the first two - so don't do both of them! Problem 1 is database focused, problem 2 is machine learning focused. I would recommend doing problem 1 only if that is a topic that interest you specifically.

The hand-in shall be in form of a report, the code used for the answers (either as Jupyter notebooks or Python code files, or as a GitHub site that I can clone), and the database in problem 1 if you decide to do that (again either on a GitHub site or made available for download somewhere). For the report, please be careful in justifying your choices of method/parameters/approaches - for the grading the quantitative answers and factual correctness of your solution will count 60% and the clarity of the report and in particular the justification of your approach will make up the remainder of the grade. Here I specifically mean that I do not want a statement that you chose method X because it was easy to implement - justify the choice on the basis of interpretability of the model, or efficiency of the approach, or speed of execution etc - basically anything that has a scientific reasoning behind.

The quality/readability of your code will **not** be graded, but if the code is readable it will make it possible for me to understand whether the wrong answer you got was a simple mistake or a fundamental misunderstanding which would affect the grading.

The reuse of code provided on the GitHub site or in the lecture notes as a starting point for your code is strongly **encouraged** but not in any way required. Please acknowledge this use in your write-up, but use thereof has absolutely no influence on your grade. You are also allowed to make use of small code-snippets and information from the web as long as this information is a) permitted to be re-used and b) that you have clearly acknowledged its use in the code. However the final code must be clearly your work and code from the web should make up less than ~10% of the final code.

Additional instructions about the questions will be given below so read carefully through the problem set.

1. A database for a time-domain survey

The last decade has seen a steady increase in the importance of time-domain astronomy and this is set to continue into the current decade with massive new facilities such as the LSST survey on the Rubin telescopes and robotic survey facilities such as the Zwicky Transient Factory having recently been or soon to be active. In this question you will design a database to hold data from such a facility, specifically it will base itself loosely on the The Vista Variables in the Vía Láctea Survey (VVV, https://vvvsurvey.org/) - no knowledge of this survey is needed at all and the survey I describe below differs from VVV in several important ways.

The question both asks you to design the database and to ingest some test data into it, read through the whole question before starting to answer it.

Survey description:

The survey for which you will design a database is carried out by taking images of a patch of the sky at fairly regular intervals. Every part of the survey area is imaged once using five filters: Z, Y, J, H and Ks. After this, the survey continues to re-observe the same patch of sky in the Ks band to build up light-curves of stars. The number of observations of a particular sub-area varies from 50 to 300.

Every time an observation is done, the stars in the field are detected and the flux of the stars in three differently sized apertures is calculated. The quantities recorded for each star is:

```
RunningID - A counter for the objects detected in a given image. This can
not be compared between images (catalogues).

Ra        - The right ascension of the object in decimal degrees.

Dec       - The declination of the object in decimal degrees.

X         - The x location of the object in the image in pixels.

Y         - The y location of the object in the image in pixels.

Flux1     - The flux of the object in counts in an aperture with 1" diamet-
er.

dFlux1    - The uncertainty on the above.

Flux2     - The same as Flux1 but with an aperture of sqrt(2)" diameter.

dFlux2    - The uncertainty on the above.

Flux3     - The same as Flux1 but with an aperture of 2" diameter.

dFlux3    - The uncertainty on the above.

Class     - The type of object: -1 if star, 0 noise, +1 if non-stellar, -2
if borderline stellar.

Mag1, Mag2, Mag3 - The calibrated magnitude of a star in the same apertures
as used for Flux1, Flux2 and Flux3.

dMag1, dMag2, dMag3 - The uncertainty on the above.

StarID    - An integer identifying a particular star. This can be compared
across catalogues.
```

Data for three different fields and observations in multiple filters, including multiple observations in Ks of some fields, is provided with the problem set. See the README file for details. Note that the files are not specifically designed for a particular database layout and you are encouraged to transform the files if that suits your layout decisions, and even create new files if needed. However it must be

done in a predictable manner - so that if you got another batch of files they could be automatically processed.

Added comments:

- Whenever colours are requested/used below, you can use just aperture #1.
- The different epochs of Ks observations are ordered such that E001 is before E002 etc.
- The MJD and the date in the file-names in the information file `file_info_for_problem.csv`, do not correspond. Use the explicit MJD values and ignore the date in the filename.

a) **Creation of a schema for the database.**

The survey manager wants to have a database that can keep track of the location of the reduced images, the catalogues of detections in each image, and she also wants to keep track of the colours of the stars and the variability of the stars in the Ks band. To make sure that the database meets the scientific needs of the team she also collated a few sample queries from the team, listed as R1-R5

R1: Find all images observed between MJD=56800 and MJD=57300 and give me the number of stars detected with S/N > 5 in each image.

R2: Find the objects that have J-H > 1.5.

R3: Find the objects where Ks differs by more than 20 times the flux uncertainty from the mean flux.

R4: Find all catalogues that exist for a given field.

R5: For a given field I would like to retrieve the Y, Z, J, H and Ks magnitudes for all stars with S/N > 30 in Y, Z, J, H and Ks.

Explain how you would lay out this database and write down a schema for the tables you think you need. There are ambiguities in the constraints listed, point these out and make a specific choice and explain your reasoning.

b) Devise SQL queries that can answer R1, R2, R3, R4, and R5. The answers should preferentially be done fully in SQL and not rely on Python but sometimes you might have opted for a database design that does not allow for this. This might be completely ok but please justify this decision. Ingest the data provided and provide the result of the queries. If the results consists of more than 20 entries, please provide the results graphically.

c) The Euclid space mission will have three filters: Y, J, and H. For planning purposes a colleague asks you to use your database to create simulated stars for Euclid. Use the distribution of your sample of star in Y-J, J-H space to create a new sample of 100,000 stars for him to use as simulation sample for Euclid. Make sure to carefully justify the choices you make. Display the result as a 2D distribution function in the Y-J, J-H plane.

Note: Y, J, H means the magnitudes, not fluxes, - take Mag1 etc as your reference.

2. Photometric redshifts of galaxies

The distance to a galaxy can be a challenging measurement to make. For most extra-galactic objects, we can not get true distances, and rely on redshifts. As long as the galaxy is sufficiently far away, this is very close to the true distance and we will here focus on redshifts.

Redshifts can best be determined using spectroscopy, but for large samples this requires too much observing time to be practical. For that reason many current and future surveys depend on the possibility to determine distances using photometry. These are known as photometric redshifts and they are the focus of this problem.

For our purposes this can be considered as a process where we learn a function $f(\theta)$ that approximately predicts the redshifts of a galaxy when given parameters θ.

There are two files that come with the project in the directory Datafiles on the Github site. These are memorably called PhotoZFileA.vot and PhotoZFileB.vot, I will refer to these as file A and file B respectively below.. These are VOtables and contain the following information:

```
Counter:  Just a running counter, starting at 1
mag_r: The r-band total magnitude of the galaxy.
u-g: the u-g colour of the galaxy
g-r: the g-r colour of the galaxy
r-i: the r-i colour of the galaxy
i-z: the i-z colour of the galaxy
z_spec: the spectroscopic redshift of the galaxy. We will take this as the
true redshift.
```

The requirements of large future cosmological surveys are that $\left\langle \left| (z_{\text{phot}} - z_{\text{spec}})/(1 + z_{\text{spec}}) \right| \right\rangle < 0.01$ or even smaller. To be consistent with this, use

$$E(\theta) = \text{median}(\left| (z_{\text{spec}} - f(\theta))/(1 + z_{\text{spec}}) \right|)$$

as the measure of the discrepancy between photometric and spectroscopic redshifts.

a) Explain what feature extraction is and what considerations you should make when carrying it out for a classification/fitting task.

b) The early work on photometric redshifts was done using linear regression. Design a regression estimator using either ridge, LASSO or linear regression to predict photometric redshifts, make sure to justify all your choices you make. Use the whole of file A for training and aim to obtain $E(\theta) < 0.01$ as training error.

c) Explain why the error estimated on the training sample is not a reliable estimate of the generalisation error of an estimator. Use file B to quantify the generalisation error of the estimator you derived in problem b.

d) Implement a photo-z estimator using the method of your choice from any of the regression methods discussed in the course, excluding the linear regression methods, e.g. k-nearest neighbours, random forests, neural networks, boosting methods, but aim to get a lower generalisation error than your method derived in problem b.

e) Discuss the advantages/disadvantages of your chosen method relative to the linear regression methods used in problem b).

3. The likelihood of gravitational wave neutron stars

The file `Datasets/pulsar_masses.vot` contains masses of pulsars from Özel & Freire 2016, Annual Reviews of Astronomy and Astrophysics, taken from the associated web page.

On October 15, 2017, the LIGO/Virgo consortium and host of ground- and space-based observatories announced the discovery of an optical counter-part to the gravitational wave source GW170817. The two neutron stars that are thought have merged to produce the gravitational wave signal, have estimated masses of $M_1 = 1.81\,\mathrm{M}_\odot$ and $M_2 = 1.11\,\mathrm{M}_\odot$.

a) Use the pulsar mass catalogue to estimate the likelihood of finding a neutron star with $M > 1.8\,\mathrm{M}_\odot$.

b) Actually the mass estimates are ranges, and the masses were $M_1 \in [1.36,2.26]$ and $M_2 \in [0.86,1.36]$. What are the likelihoods of those mass ranges and the likelihood of the binary?

c) Simulate the next 5 detections of merging pulsars with LIGO+Virgo. What is your prediction for the average mass neutron star they would detect? [hint: to draw N from a `KernelDensity` object, you can use the `.sample(N)` function]

d) There are many assumptions made that I did not spell out - make a list of some that you worry about.

4. How many peaks?

The file `Datasets/mysterious-peaks.fits` contains data drawn from a distribution with multiple peaks. How many peaks are there?

It is a simple fits file that can be read in with `astropy.io.fits`