Job Interview Task (Estimated time 6h)

The final goal of the task is, with the provided dataset, to build an ML prediction service.

1.1 Data analysis and feature engineering

Data exploratory analysis. Get a general picture of the data. Identify problems within the data, and propose and apply feature engineering solutions. These analyses should help you to decide on how to proceed with the next task.

1.2 Modelling

Once the data is ready, train a classifier. You are free to choose the type of model but, please, justify your decision. How you train your model and which metrics you are evaluating are important.

1.3 Dockerising, ML as a service.

Save the model in a binary file and, using a Flask in a Docker container, create a REST API. For a well-formed prediction request, the Flask app shall load the model, compute the prediction and will respond to the prediction request with a JSON.

1.4 Databases and UI

When there's a prediction made, the app shall store: the current timestamp, the input values and the prediction in a database (not in a file). Additionally, the Flask app should have a URL destination in which we can see a simple table with all the saved predictions. You are free to choose the type of database but, please, justify your decision.

Hints:

- *Write down what you are trying to achieve at every relevant step.*
- *Comment out your results.*
- *Always justify your choices.*
- *Comments on how you would improve any part of the project are more than welcome.*
- *Functionality is preferred over beauty. Do not waste time on making the app look prettier unless your API, your data analysis, and your model are perfect.*
- *You are free to choose the type of model, but some models tell us more about their predictions than others. The most informative ones are better valued.*

Please, deliver the source code and the docker commands to build and run your containerized application; do not send us the docker image.

Variables description:

objid = Object Identifier

ra = J2000 Right Ascension (r-band)

dec = J2000 Declination (r-band)

u = better of deV/Exp magnitude fit (u-band)

g = better of deV/Exp magnitude fit (g-band)

r = better of deV/Exp magnitude fit (r-band)

i = better of deV/Exp magnitude fit (i-band)

z = better of deV/Exp magnitude fit (z-band)

run = Run Number

rerun = Rerun Number

camcol = Camera column

field = Field number

specobjid = Object Identifier

redshift = Final Redshift

plate = plate number

mjd = MJD of observation

fiberid = fiberID

class = object class (galaxy, star, qos [Quasi-Stellar Objects])