CS342 Assignment 2

Callum Marvell 1606521

November 18, 2018

1 Task 1

Thanks to the information presented here [2], histograms detailing the distribution of time-series lengths and the time at which each image was taken by the LSST (via use of mjd) were plotted to get a basic idea of the qualities inherent to the time-series data. This mainly revealed two things:

- That the length of the time-series was highly variable (so assuming that they are all the same length is not sensible) and that they will require considerable work to be converted so that all are the same length.
- That the images taken by the LSST were not taken at a constant rate throughout - many were taken during a few "spikes" and several "troughs" exist where very few were taken. This must be taken into account before making any assumptions abou the data gathering methods and consistency.

Following on from this, several other graphs were created to investigate the distributions of various recorded statistics. These include histograms of the distributions of `hostgal_specz` and `hostgal_photoz` and a scatter graph of `ra` against `decl`.

2 Task 2

3 Task 3

4 Task 4

The first attempt made at building a RandomForestClassifier to correctly classify the objects was not very successful - only scoring around 0.6 in R^2 score. This was likely due to the significant reduction in usable data, as a direct result of my quick and dirty workaround for NaN results in the dataframe (removing any columns with any NaN results inside). The next attempt switched this concept around - instead only removing any rows which contained NaN results (from both the training and test set). Additionally, an attempt to use the `class_weights` determined here [1] via frequency analysis, in the hopes of improving the score of the classifier. The move to using `log_loss` as in the leaderboards was also made.

5 Task 5

6 Task 6

7 Task 7

References

[1] Public class weights. `https://www.kaggle.com/c/PLAsTiCC-2018/discussion/67194`. Accessed: 18-11-2018.

[2] Strategies for flux time-series preprocessing. `https://www.kaggle.com/mithrillion/strategies-for-flux-time-series-preprocessing`. Accessed: 12-11-2018.