Databases and Datamining in Astronomy Final Project Report

Louis Martin
s1886134

January 26, 2018

1 Report Layout & Introduction

The emphasis of this report is to show my understanding of what was taught in the *Databases and Datamining in Astronomy* (DBDMiA) course by ensuring that I can reason my choices of models and decisions when tackling problems. The first question looks at database creation and manipulation through SQL and python and then focuses on the Datamining aspect of the course in the second question. The report will be set up as follows with each section being addressed as a subsection of the following titles *Methods*, *Results* and *Discussion*. Finally an over-all conclusion will look at the both questions commenting on what I had done as a wrap up. All codes are found on found on the github page https://github.com/LouisMartin93/Final_Exam_DBDMiA.

With the introduction of larger and larger memory capabilities on computers, more and more data has been gathered than ever before. While storing this huge amount of information is all good and well, organising and accessing it in a predictable and easy manner is key to allow an understanding from these large volumes of data. This is were the introduction of databases is needed. Databases can range from a variety of topics and are widely used in astronomy. The increase in storage space of data has allowed for extensive multi-observation surveys looking variability and transits in night sky examples of this are Vista Variables in Vía Láctea Survey or the MASCARA survey and not to mention the 2MASS and SDSS catalogues are all huge databases containing a number of properties on stars. Generally these databases will contain multiple tables linked to one another to allow understanding of the relation of characteristics of the table entries be it stars, galaxies etc... specific languages are used to link and query results from these tables to take samples or specific selections (SDSS has information on 260,562,744 star [1]). The most commonly used language is the Structured Query Language (*SQL*), which allows for the creation of new tables, databases, selecting specific entries and outputting relevant information. But how does one draw conclusions and correlation between information from these databases? This is were data mining comes in.

Data mining can be considered the accumulation of the following topics Machine learning and pattern-recognition using a pre-processed database as a source of data. Machine Learning is described as, Machine learning is essentially a form of applied statistics with

increased emphasis on the use of computers to statistically estimated complicated functions and a decreased emphasis on providing confidence intervals around these functions [2]. From this definition it is easy to associate Machine Learning (ML) as a method of pattern reignition. Datamining is typically broken up in to 4 categorises: Classification, Estimation/prediction, Grouping and Dimension reduction. Examples of where these are used in astronomy are photometric redshifts of galaxies (prediction), galaxy morphology (classification) and Identifying asteroids from photometric data (grouping). The focus of this report with be on the category of Predictions using the methods of regression estimators, Gaussian Mixture Models and K-Nearest Neighbours to predict values of distributions and find relations between characteristics (features) of galaxies. Use of these techniques gives the science community a method of predicting values for models and making classifications robustly and autonomously, making for a more efficient system of operation.

2 Methods

2.1 Question 1 - A Database for a Time-Domain Survey

Here the data of several survey images for 3 different fields in 5 different filters are used to create a database of stars. In each image 10000 objects are detected but are not necessarily classified as stars so a classification is set for each object by the survey coordinators. The Task at hand is to create a Database which is robust enough that it can be recreated if another set of files are added to it and an example set of SQL queries can be preformed.The code used for the answers to the following questions can be found in ipython notebook entitled *'Q1-Exam.ipynb'*.

For the creation of the data base in *Q1 a)* use of the python packages *sqlite3*, *pandas* and *astropy* made the creation of these tables easy and compact. The database was called *Final_DB.db* and the tables were named as *info_table* and *Object_info*.

SQL commands were used to create tables which if entries exceeded 20 rows would be displayed graphically- usually through histograms with a shared y-axis. Multiple histograms would be used when dealing with distributions over multiple fields (R2) or multiple filters (R5). The SQL commands are interpreted using the python package *pandas* and can be found in the afore mentioned ipython notebook.

A Gaussian mixture model of 12 gaussians from the python package *sklearn* was used to determine the distribution of Y-J vs J-H stars from the database created in a). A new table of starID, FieldID as well as Y-J and J-H values was first created with SQL queries so manipulation of the data was easier. Fitting the model to the data, a sample of 100000 was generated as a simulated sample for the Euclid mission. GMM works on the principle that the distribution can be written as a sum of Gaussians:

$$P(x_i|\theta) = \sum_{j=1}^{M} \alpha_j N(\mu_j, \sigma_j) \tag{1}$$

where P is the probability M is the number of components μ_j and σ_j are parameters of the

gaussian j and α_j is the jth co-efficient. Equ. 1 is then written as the the log likelihood and solved using an Expectation-Maximisation algorithm:

$$lnL = \sum_{i=1}^{N} ln\Big[P(x_i|\theta)\Big] \tag{2}$$

This is all implemented in the *sklearn* package.

2.2 Question 2 - Photometric redshifts of Galaxies

For *Q2 b)* the assignment asks to use a regression method to make fitted model from the data to get an error better than 0.1. The error was calculated as the following:

$$E(\theta) = median\Big(\Big|\frac{(z_{spec} - f(\theta))}{(1 + z_{spec})}\Big|\Big) < 0.1 \tag{3}$$

Where $f(\theta)$ is a fit function and z_{spec} is the galaxy red shift read in from the data files. There are 3 types of Regression algorithms: Linear, Ridge and Lasso regression. The three methods are described by minimising the following equations:

$$RSS = \sum_{i}(y_i - f(x_i))^2 \tag{4}$$

$$\text{Where } f(x_i) = \theta_0 + \sum_{j=1}^{p}(\theta_j x_{ij}) \tag{5}$$

$$RSS + \lambda \sum_{j=1}^{p} \beta^2 \quad \text{(Ridge)} \tag{6}$$

$$RSS + \lambda \sum_{j=1}^{p} |\theta_j| \quad \text{(LASSO)} \tag{7}$$

Where λ is the regularisation parameter and β is the parameter vector. For this particular case, a Ridge regression [3] method (equ.6) was used to fit training data in file 'PhotoZ-FileA.vot'(file A) to a model. This model is then used in *c)* to evaluate a generalisation error of the model on data that had not been used to create the model in the first place ,'PhotoZfileB.vot' (file B).

In *d)* an alternative method is required for making the model to further improve the generalisation error. For this requirement an Ensemble method of Bagging (**B**ootstrap **Agg**regat**ing**) [4] accompanied with a K Nearest Neighbours (KNN) [5] regression method was chosen. The Idea behind the KNN approach is simple, for every point the mean of the K nearest neighbours are taken with an option to give inverse weighting based on distance from the chosen point. Bagging works on the principle of averaging fits from m sub-samples of the the training data to create a smoothed output in the case of Regression. Implementations of these are found in the ipython notebook *'Exam-Q2.ipynb'*.

3 Results

3.1 Question 1 - A Database for a Time-Domain Survey

Below in tables 1 and 2 are the schema's of the two tables in the databases. This was constructed based on the requirements of queries (R1-R5) in the next section. These queries have some ambiguities which effect the interruption and thus the SQL command and output. These ambiguities are discussed in section 4.

Column	Type	SQL Type	Other
Filename	String	VARCHAR(36)	Primarykey, Unique
Filter	String	VARCHAR(2)	Foreign
MJD	Float	FLOAT	Foreign
FieldID	Integer	INT	Foreign
Airmass	Float	FLOAT	-
Exptime	Float	FLOAT	-

Table 1: This is the Schema layout for *info_table*. SQL syntax can be found in *Q1 - Exam.ipynb*

Column	Type	SQL Type	Other
StarID	Integer	INT	Primary key
X	float	FLOAT	-
Y	Float	FLOAT	-
RA	Real Number	DOUBLE	-
DEC	Real Number	DOUBLE	-
Flux1	Float	FLOAT	-
dFlux1	Float	FLOAT	-
Mag1	Float	FLOAT	-
dMag1	Float	FLOAT	-
MJD	Float	FLOAT	Foreign
Filter	String	VARCHAR(2)	Foreign
FieldID	Integer	INT	Foreign

Table 2: This is the Schema layout for *Object_info*. SQL syntax can be found in *Q1 - Exam.ipynb*

The results from the SQL commands can be seen in the form of tables and histograms below. Each showing either a list of specific values or a large distributions given field or magnitude. Tables 3,4 and 5 answer the queries R1, R3 and R4 where as Fig.1 and Fig.2 show histograms displaying the information to describe query R2 and R5. Tables and plots are in order of Query from R1-R5

Fname	FieldID	Filter	MJD	num
H-ADP.2017-01-18T11:58:35.780.fits	1	H	57257.044108	7982
H-ADP.2017-01-18T11:58:35.780b.fits	2	H	57258.044108	7725
H-ADP.2017-01-18T11:58:35.780c.fits	3	H	57258.044108	8022
J-ADP.2017-01-18T11:58:35.781.fits	1	J	57257.050432	7022
J-ADP.2017-01-18T11:58:35.781b.fits	2	J	57258.050432	7354
J-ADP.2017-01-18T11:58:35.781c.fits	3	J	57258.050432	7248
Ks-ADP.2016-05-25T15:33:43.377.fits	1	Ks	56829.039051	7888
Y-ADP.2017-01-18T11:58:36.901.fits	1	Y	57267.159665	6806
Y-ADP.2017-01-18T11:58:36.901b.fits	2	Y	57268.159665	7215
Y-ADP.2017-01-18T11:58:36.901c.fits	3	Y	57268.159665	7186
Z-ADP.2017-01-18T11:58:36.905.fits	1	Z	57267.167107	6477
Z-ADP.2017-01-18T11:58:36.905b.fits	2	Z	57268.167107	6929
Z-ADP.2017-01-18T11:58:36.905c.fits	3	Z	57268.167107	6741

Table 3: Answer to R1

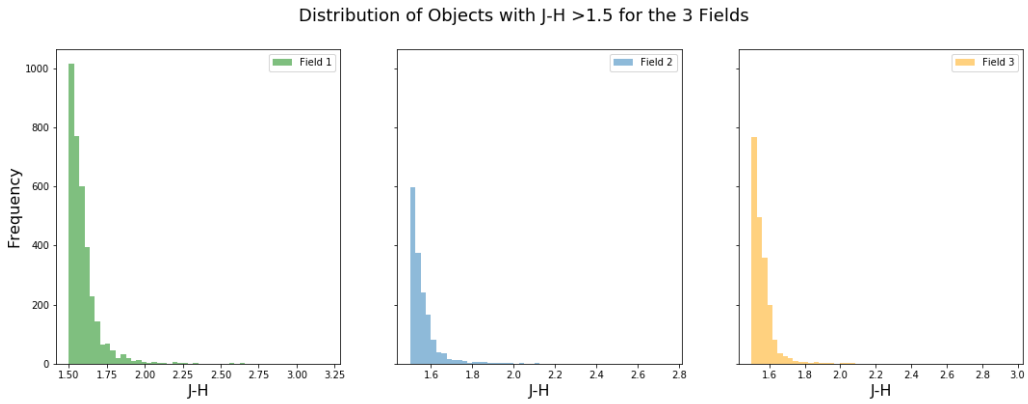


Figure 1: The distribution for J-H $\geq$ 1.5 in all fields and the answer to R2

FieldID	Number
1	2419
2	0
3	435

Table 4: Answer to R3 for all fields showing the number of objects in a given field which for all epoch differ from the mean flux of the object by 20 times the uncertainty. The Zero in field 2 indicates that there was only one epoch in the Ks filter so the variability of the objects could not be calculated.

File_Name	Filter	FieldID
H-ADP.2017-01-18T11:58:35.780.fits	H	1
J-ADP.2017-01-18T11:58:35.781.fits	J	1
Ks-ADP.2016-05-25T15:33:39.546.fits	Ks	1
Ks-ADP.2016-05-25T15:33:43.377.fits	Ks	1
Ks-ADP.2017-01-18T11:58:39.907.fits	Ks	1
Y-ADP.2017-01-18T11:58:36.901.fits	Y	1
Z-ADP.2017-01-18T11:58:36.905.fits	Z	1

Table 5: Answer to Query R4 showing the unique catalogues and associated filter (not required) for field 1. The multiple instances of Ks indicate that for this field there was more than one epoch so that variability of the objects could be measured.

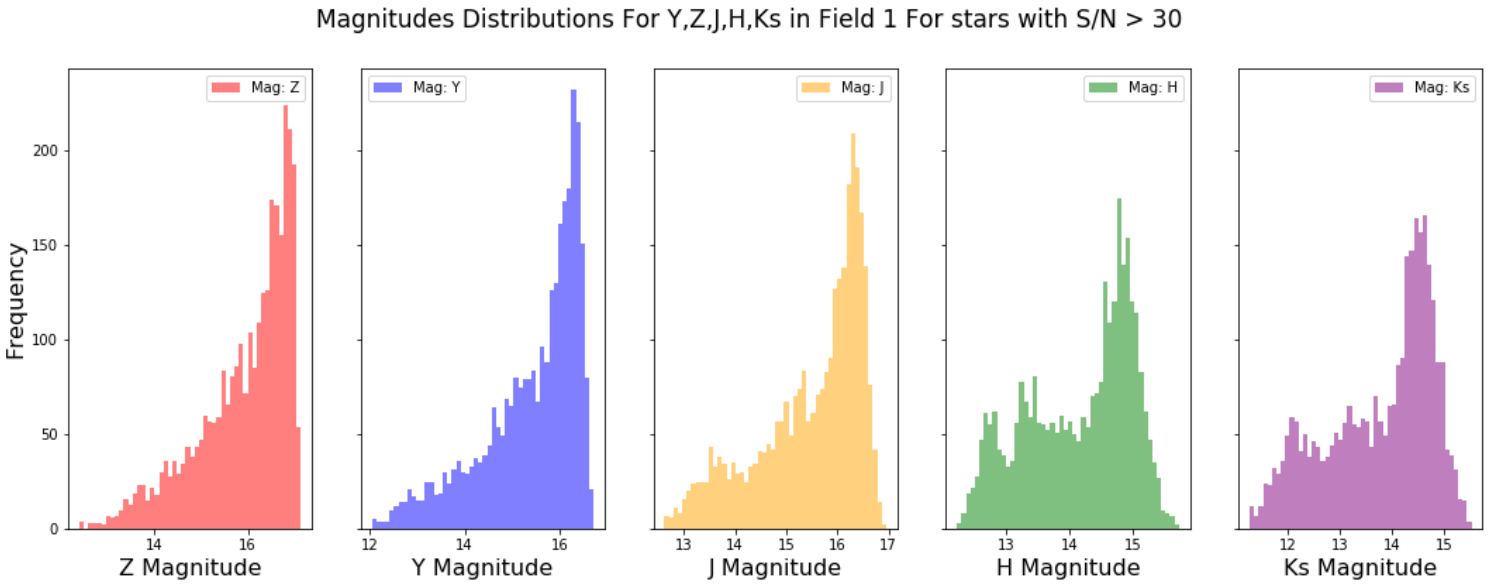


Figure 2: Distribution of stars in Field 1 with a Signal to Noise ≥ 30. This is the answer to R5. It is interesting to note that as the wavelength of the filter increase, more and more stars begin to have lower magnitudes, meaning that a lot of the stars observed are strong emitters in the infrared regime. It should be noted that the histograms are ordered in ascending wavelength

Finally the plots of of the distribution of stars given its Y-J and J-H values for each field are seen in Fig.3. Fitting Gaussian Mixture models to this data gives a model which is used to create a sample of 100000 stars based on the relation of Y-J and J-H in the 3 fields. Plots of these distributions and models are seen in Fig.4

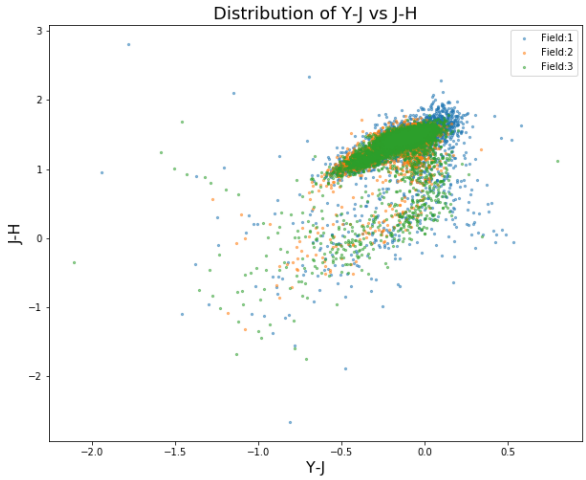
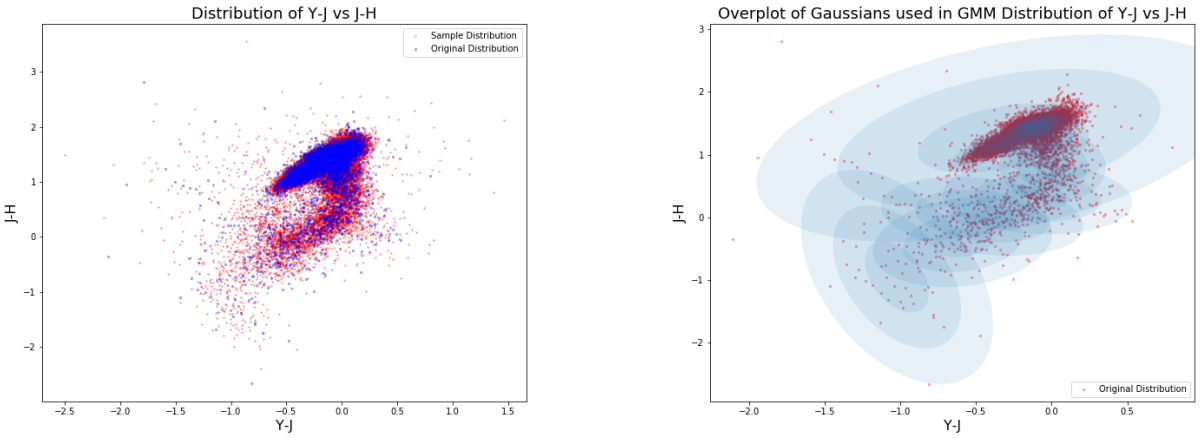


Figure 3: The distribution star for Y-J vs J-H in each field. It appears that in field 1 and 3 there is more scatter



a) b)

Figure 4: a) The original distribution as well as the new sampled distribution of 100000 stars for the Euclids mission. b) The gaussians used to calculate the distribution. For visualisation the weighting of gaussians is the same.

3.2 Question 2 - Photometric redshifts of Galaxies

For the Ridge regression to preform to the highest order and give the best fit to the data, an optimization of the regularisation parameter λ must be made. The optimal λ was validated through 10-fold cross-validation on the training data (file A) and results of this can be seen in Fig.5

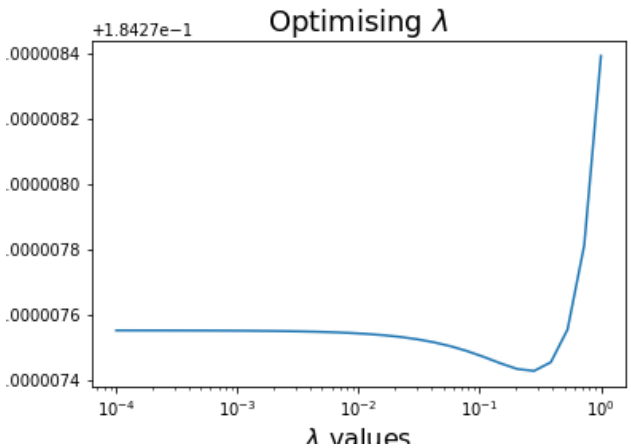


Figure 5: Plot of the cross validation (cv) values used to calculate the optimal regularisation parameter (seen at the minimum of the curve).

The optimal λ was then used to make the fit using all of file A giving the training error and file B was then used to calculate the generalisation error, both are seen in table.6 as well as a plot of the relative residuals vs z_{spec} in Fig.6

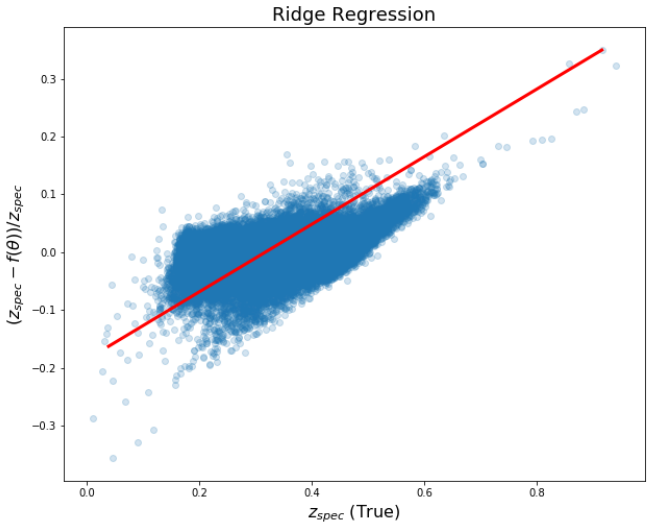


Figure 6: The Residuals of the fitted models using the test data from file B as well a median of the first and last 700 points.

For the KNN regression, the number of neighbours used must be systematically determined. This is done via 10-fold Cross-validation carried out using the training data to achieve an optimal k-neighbours of 29 as seen in Fig.7. This parameter is then used with the bagged KNN process to create a model which achieves a general error less than that calculated in *2*

b). This can be seen in table.6 and the fit seen in Fig.8.

Method	training	Generalisation
Ridge	0.01455	0.01461
Bagged KNN	0.00404	0.01243

Table 6: Table comparing the errors for the training and test set of data for the given methods of regression

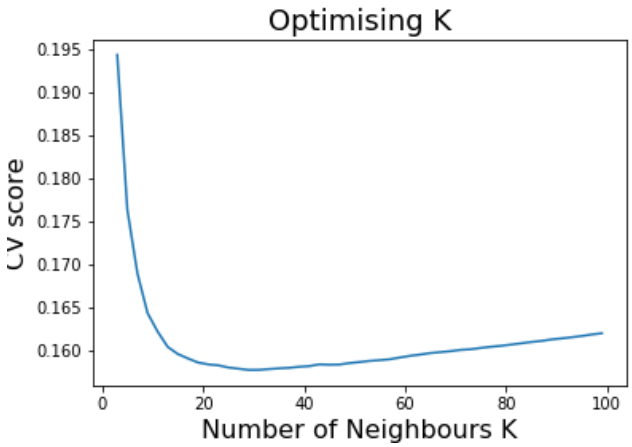


Figure 7: Plot of the cross validation (cv) values used to calculate the optimal number of nearest Neighbours (seen at the minimum of the curve.

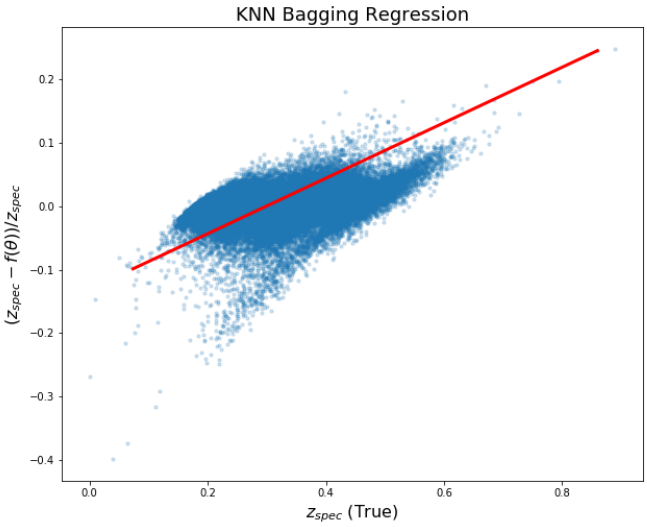


Figure 8: Residuals plot from the Bagged KNN model on the test data.

4 Discussion

4.1 Question 1 - A Database for a Time-Domain Survey

My database was created with two tables, one with the file information (*info_table*) and a single objects information table (*Object_info*). This was done so that to associate objects to a file for a given field, filter and MJD. The construction of these tables and database relies heavily on the naming convention used for the fits files (images/catalogues). One of the draw backs of this table is that as the Epochs and Fields increase the table will become very large as some information is repeated (StarID, x,y, Ra, Dec etc..) however this is needed as the use of any combination of 2 of the following FieldID, Filter or MJD will give incorrect results as these values are not unique, (fields 2 and 3 have the exact same MJDs for all Filters where there isn't more than 1 epoch) hence only using all three can files be identified. Python was also used for the construction of the tables and databases as there was no point using the schema and formatting of SQL as *sqlite* does handle schemas [6] and it was not required to create the database in SQL.

As mentioned, interpretation of each of the Queries can vary due to the outlined ambiguities below. Each query will be discussed individually, whether there was an ambiguity in the interpretation or not.

R1:
There were no ambiguities here as to what was required as it was just a counting of all the stars in the images that were taken between the given times and above a SNR of 5.

R2:
Here there was only the question of what field do I need to look at or do I need to look at all fields. Also why all objects and not just stars, as including the J-H value for noise and non-stellar (galactic?) objects seemed pointless, however it may be possible that choosing a threshold mitigates that need, but again not all stars vary a lot from filter to filter.

R3:
Personally this Query was the most vague. The *mean flux* could have meant any one of the following: the flux you are given for that star, mean of Ks flux over all fields, the mean Ks flux for a given field, the mean Ks flux for a given star. Additionally based on the problem of determining the mean flux, the uncertainty is also a bit ambiguous. However due after re-reading the assignment the star variability in the Ks band is brought in to question, so I concluded that looking at the mean flux of each star across all epochs was needed and then if the star satisfied the criteria for each epoch it would be considered for selection.

R4:
On the initial read the only confusing part of this query was the term 'catalogues' as to my interpretation that meant comparing the stars to the likes of SDSS or 2MASS. However reading of the assignment it clarifies that catalogues in this case mean the .fits images.

R5:
Determining the Ks filter to use for a given field the only unclear point as all the fields have Ks filters but some were take a different epochs so for the command that was decided on, the Average Ks for each star was taken for a single field and used when displaying the information, similarly the average SNR was taken for a each star for a given field to fulfil

the conditioning.

For the results of R1 and R4 there are no real surprises everything is as expected. Interestingly enough for R5 of the 10000 objects per field less than a third met the criteria stated in the queries, however the trends of the distributions would match theory as as the observed wavelength increased, more stars were visible at lower magnitudes. Initially there were some concerns that for R2 as to whether or not the magnitudes would need to be normalised to the same scale (as the filters were taken for different exposure times) however as stated in the data information sheet this had already been done.

For R3 and R5 with the way I had structured my database I had to resort to python to make the command generation robust enough so that if a new set of data was given for a given field it would still produce the correct command no matter how many epochs of Ks there are (assuming that Ks only will have multiple epochs). Initially a first 'table' was created resulting in the unique MJD values that were associated with the Ks filter in a given field. My command of SQL was not good enough such that I could implement a for loop of some kind in to the selection to process, hence my reasoning to use python.

A 12 component Gaussian Mixture Model (GMM) was used for creating a sample distribution of 100000 stars based on the $\approx$ 20000 or so stars that are in my database over the 3 fields for the colours *Y-J* and *J-H*. The GMM was chosen over other distributions methods such Kernel Density Estimators (KDE) due to its less computational time (finding the optimal bandwidth) and the smoothness of the distribution model (different kernels lead to different smoothness of the distributions). While these are minor issues the method is simple and efficient at approximating complex functions (such as this distribution of stars). The number of components for the GMM was determined by comparison of the a BIC and AIC tests which is calculated based on the following equations:

$$AIC = 2k - 2\ln L_{max} \tag{8}$$

$$BIC = k\ln n - 2\ln L_{max} \tag{9}$$

where k is the number of free parameters to estimate, n is the number of data points used in the fit and L_{max} is the maximum likelihood. This gave different values when comparing the minimum number of components to use in the GMM, so a mean of the two minimum of the different models was used, comparison of this can be seen in Fig.9. Using this GMM a sample of 100000 stars is drawn and results of this sample can be seen over plotted the original distribution (Fig.4).

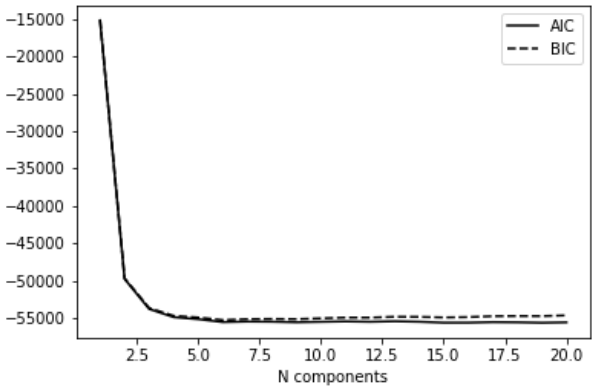


Figure 9: AIC and BIC tests calculated for the minimum number of components to be used in the Gaussian Mixture Model. A rounded Mean of AIC=19 and BIC=6 was used for the mixture model.

4.2 Question 2 - Photometric redshifts of Galaxies

Feature extraction is defined as examining the characteristics of data points and focusing on the characteristic which influence a fit or characterisation prediction of a data point. For fitting a model or a characterisation the problem of over fitting due large numbers of variables can be minimised by feature extraction. This can be done using Principle Component Analysis (PCA). This is helpful in reducing computation times, as the number of variables are reduced. In this section the variables for each galaxy such as r_mag, r-i and g-r would be potentially more influential than i-z or u-g. By doing a PCA decomposition of the data we can show which are the most influential variables and which values could be combined to reduce the dimensionality.

The choice to run Ridge regression to model the fit function for z_{phot} was based the assumed correlation between the parameters such as the r magnitude and the colour difference g-r and r-i. Because of this correlation (if high) there would be a high variance if only linear regression was used to calculate the fit. To minimise the variance Ridge regression introduces a bias through the $\lambda \sum_{j=1}^{p} \beta^2$ term in equ.6, however this will give a better fit for the multi-correlated data, which is determined through the pair plot of fig.10.

As posed in *Q2 c)*, using the training error as the general error is invalid as the the training error represents the error of the best fitted model for that particular set of data used to train it. A general error could be evaluated as the error of the model (as derived in 2b) using data that has not been used to derive the model in the first place avoiding bias between the model predictions and actual values as well as showing the robustness of the model.

The Bagging process was chosen as it is known to improve the stability and accuracy of machine learning algorithms used in regression as well as reduces variance and helps to avoid over-fitting, however the later is not of a concern to model used in this report as the

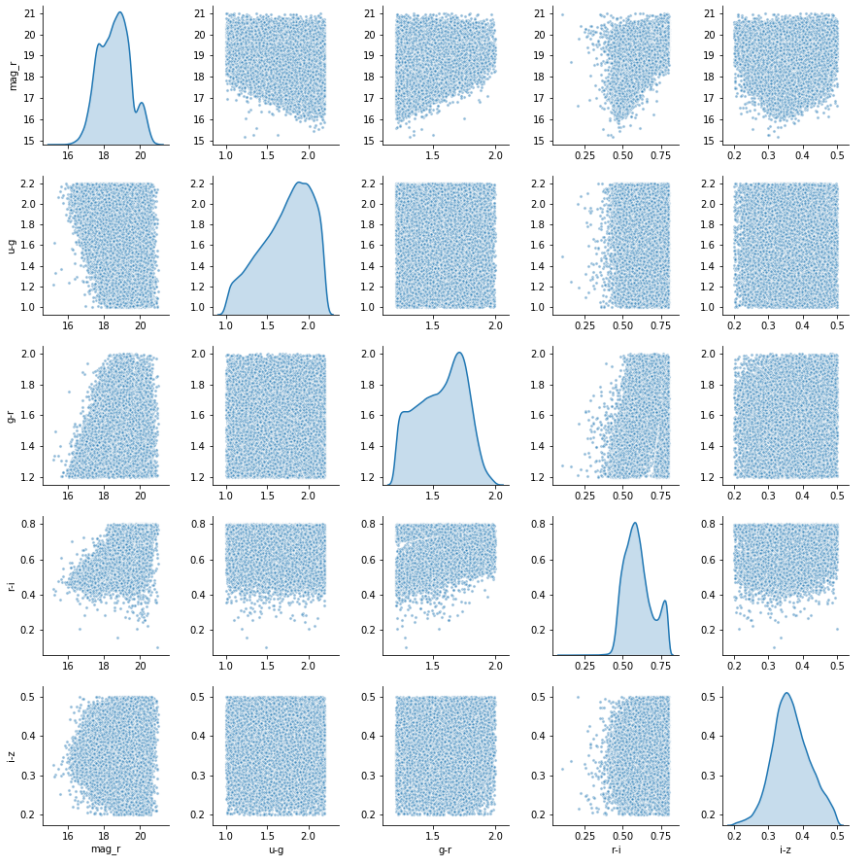


Figure 10: This pair plot shows the relations of the features with one another as can be seen mag_r has a slight correlation with all the colour difference, dues to it being used to calculate the values of the magnitudes in 'g-r' and 'r-i'

fitting follows a KNN regression fit(1st order multi-variable polynomial).

The main advantages of the bagged KNN regression model over the Ridge linear Regression model is the use of averaging of the bootstrapping method to smooth the fit as well as the weighting of points that can be used in the KNN method make the fit more accurate where as the Ridge regression tries to fit all points to the one fit based on uniform weighting.

Another advantage of Bagged KNN is its general adaptability, as this form of machine learning can also be used in classifying and grouping of points, compared to Ridge Regression which is as the title suggests is confined to regression problems.

In the case of Feature extraction and dimension reduction, using PCA on the data before applying KNN will isolate highly contributing feature and use them for the regression fit, this is particularly used full in high dimensional data and data with redundant information. Similarities of the two methods include their reduction of the variance of the model.

The separate approach doesn't come with its disadvantages either, as computational time becomes a factor when dealing with large data sets of high dimensional order (variables) and having to verify the optimal number of neighbours (K) as well as running the algorithm for higher numbers of estimators can be come time consuming. As a comparison of computation time can be seen below with. A concluding remark on this is that the while the general error

may be lower in the case of the bagged KNN compared with the ridge regression method, it still fails to meet the requirement of a model for photometric redshifts (see equ.3). Something to be looked at is the time taken to achieve these better results worth it.

5 Conclusion

Initially the database was to be set up as 4 tables, a file information table and a table for each field, however this ran in to a similar problem of storage (however not as severe) as the class of an object in each filter for a given field was not necessarily the same therefore custom columns of the flux and magnitude for each filter could not be created (this was the solution to limiting the number of entries). Just as the database layout changed so too did the table schema's. MJD was not to be a column in the *Object_info* table as it was only realised later that the MJD was the same for field 2 and 3. This fact made me question as to whether this was a mistake or if stars in field 2 and 3 were actually the same, but StarID settled that argument and so assumed it was a mistake unless of course there were multiple cameras on the one Field or Field 2 and 3 were sub-fields of a single image.

If computation time wasn't an issue use of and optimised (for the number of estimators) Random forest method for *Q2 d)* may have been more accurate approach as this also accounts for the features of the data when making the decision trees. Also using having only discover this recently it is not always wise to use Bagging techniques on KNN as this will degrade the performance of the regression method (albeit only mildly) [7].

References

[1] SDSS website http://www.sdss.org/

[2] Goodfellow, Bengio, Courville *Deep Learning* ,MIT Press, http://www.deeplearningbook.org, 2016

[3] Hoerl, Arthur E and Kennard, Robert W, *Ridge regression: Biased estimation for nonorthogonal problems*, Taylor & Francis Group,1970

[4] Breiman, L *Bagging predictors. Univ. California Technical Report No. 421*, 1994

[5] , Altman, Naomi S, *An introduction to kernel and nearest-neighbor nonparametric regression*, Taylor & Francis,The American Statistician,1992

[6] sqlite schema reference, https://pandas.pydata.org/pandas-docs/stable/io.html#schema-support

[7] L. Breiman, *"Bagging predictors"*, Machine learning,1996