university of groningen / faculty of mathematics and natural sciences / kapteyn astronomical institute

Bachelor Project Astronomy

Unsupervised feature selection in astronomical surveys

Sander Verdult
s1715909

Supervised by
Prof. dr. Reynier Peletier
Teymoor Saifollahi

August 2020

Abstract

With astronomical surveys continuing to increase in scope and ambition, more information is becoming readily available to the astronomy community. Not only do we have a greater number of objects to analyse at once, but we also have more features (parameters) available to us. However, scientific models trying to accommodate all these features at once become either overly complex or inaccurate due to overfitting. In this project, we will look into one of these astronomical surveys (GAMA) and create a catalogue of galaxy features consisting both of photometric and spectral information. After preselection using noise ratios, primary chi-square and the Extended Isolation Forest (EIF) anomaly detection algorithm, we will explain and then apply two different Unsupervised Feature Selection techniques (Principle Feature Analysis and a Hybrid algorithm) to this dataset in an attempt to define the most important features of galaxies. We shall examine the findings made, discussing lessons learned and possible steps to improve similar projects. Finally, we shall select a subset of both "best features" and a randomly selected subset, and compare results of a K-means clustering algorithm to get an indication of the viability of these techniques. In the end, we conclude that the used algorithms offer potential, but that they will require further work and modifications to provide compelling clustering results.

ACKNOWLEDGEMENT

This thesis would not have been possible without the assistance, support and motivation provided to me by numerous people. There are some I would like to thank in particular.

I would like to thank my supervisor prof. dr. Reynier Peletier.

I am most grateful for the supervision, insights and patience given by Teymoor Saifollahi, whom first suggested this project.

To my friends, especially those who were available during these crazy times to help with motivation or encourage each other into studying and not giving up. Thanks to Meike, Rogier, Dora, Giliam, Eva and Sarah in particular. You guys kept me going and provided made sure I never gave up.

And of course, my direct family, who have supported me for all these years. Thanks, Mom, Dad and Marit.

Contents

I. INTRODUCTION

One can define the current human era as one about information. This focus on data has allowed not just science, but civilisation as a whole to develop rapidly, granting humanity new insights, new ways of communication on a scale never seen before. So too within the field of astronomy. Where pioneers like Messier and Herschel once produced catalogues of 102 or up to 2500 objects, we now have surveys like GAIA[1] which have information about more than 1 billion objects.[1] With this increase in information-density came the development of new techniques to study this. We rely upon machines to compute for us and algorithms to predict models, formulae or the future. However, all these techniques require careful examination to ensure that we are using the correct methods in the proper places.

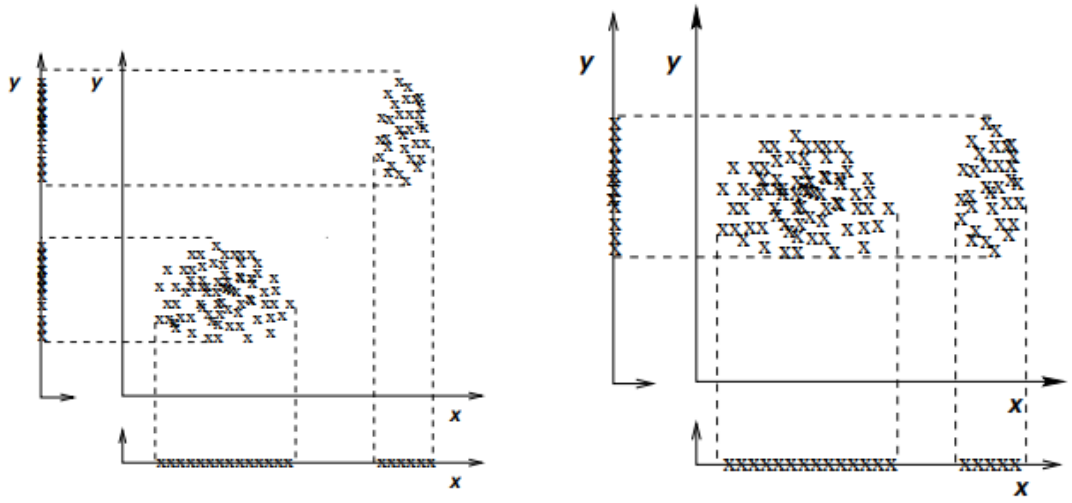
i. Machine learning

Machine learning is the study of computer algorithms which use statistics to find patterns within data. The key here is that these algorithms are not explicitly programmed to solve specific programs, but instead go through a learning process to teach themselves how solve particular problems or apply to specific datasets. Within machine learning we can distinguish three different approaches.
Supervised algorithms where labelled data is used in a training phase to create a model, after which a predictive phase occurs where the previously constructed model is used to predict something about the dataset as a whole. This prediction can be about *classification* (where the model is used to map objects into discrete output variables, i.e. putting labels onto objects) or for example about *regression* (here models are used which predict continuous values, allowing us, for example, fit data into to models or formula).
Unsupervised algorithms, on the other hand, work without labelled data. They have no "base truth" with which to compare new information. These still tend first to have a training phase, where they use random samplings of the data to create models, before applying these models to the database as a whole to make predictions. This category includes *clustering algorithms* (finding groups of similar objects), *dimensionality reduction* (algorithms which try to fit the information we desire into fewer features) and outlier detection techniques. [2]
Furthermore, there is the approach of *Reinforcement Learning*, where a computer program interacts with its environment to perform a specific goal. As it iteratively attempts different solutions to this problem, rewards get provided as feedback in cases where it performs well, allowing the program to try and maximise these

[1] the ESA astronomical observatory mission with the goal of surveying the position of 1 billion stars

rewards.

As datasets grow in feature size, a phenomenon called the curse of dimensionality comes into effect. This states that as the number of features increases, the predictive power of models decreases while the computational power required to investigate them increases. This effect has influence on for example clustering algorithms, modeling, fitting data, predictive algorithms and anomaly detection. [3]) On top of this, many datasets contain features which are either irrelevant or redundant. These features can affect the analysis of the data, making the analysis less reliable, creating biasses or even incorrect models. [4]



(a) *In this example, features x and y are redundant, because feature x provides the same information as feature y. We need to pick only one to discriminate between the two clusters*

(b) *Here feature y can be considered to be irrelevant. It's exclusion would cost us little information. Would feature x be omitted however, we can only detect one cluster.*

Figure 1: *2 dimensional examples of of redundancy and irrelevancy. When more dimensions are involved, the problem becomes harder to visualise and inspect. Hence the need for machine learning.[5]*

ii. Feature selection

Feature selection is typically used to reduce the complexity of models, improve the accuracy of models with the right subset and to reduce overfitting.[6] By finding ways to select only the most useful features, we can create better models for tasks like classification or clustering. This feature selection not only reduces the dimensionality of the data, allowing for better visualisation and understanding; it also commonly leads to more compact models with better generalisation ability.

While there are also supervised approaches to feature selection, in this project, we will not be using a labelled set of data. We shall therefore be focusing strictly on unsupervised machine learning methods. Compared to supervised methods, the field of unsupervised machine learning is currently less developed, and as far as we could determine, no applications upon astronomical data have been published so far.

The primary goal of this thesis is to use various unsupervised machine learning techniques to find *"the most important features"* with regards to galaxies observed in the Galaxy And Mass Assembly (GAMA) survey. There are multiple ways to approach unsupervised feature selection. *Filter methods* look at intrinsic properties of the data itself, and selects a subset of these properties based on some statistical test or data structure. *Wrapper methods* evaluate features using the results of a clustering algorithm. They look for subsets of features that result in better clustering results. However, for large feature sets, there are lot's of potential combinations and so the computational cost is typically high. Finally, there are the *Hybrid methods*, which combine Filter and Wrapper methods in an attempt to find a compromise between efficiency and effectiveness.

In order to find "the most important features", we should set a secondary goal. After all, the features which are most important change depending upon the goal we are trying to achieve. Here we will define this secondary goal as: *to apply clustering algorithms to a subset of features from the GAMA dataset so that we can uncover "interesting" clusters of objects* [2]
We can define "the most important features" as whatever feature subset we can select that provides "interesting" clustering result. To avoid further complications, we shall be using the same clustering method of k-means everywhere. We will be comparing the implementation and results of two different feature selection methods (Principle Feature Analysis and Solorio-Fernández et al.'s new hybrid method.) Finally, we will be using the results of these Feature selection methods to do clustering within the GAMA feature space.

Though this project was heavy on (python) programming, this thesis will not be. As a result, some snippets of code have been deferred to the appendix while most of the code will be made available on the following GitHub page: `https://github.com/Dainean/astro_UFS`. Information about used modules and how to install these will also be found here, along with some additional results in graph form, along with information on how to produce these yourself.

[2] This secondary goal is however not the focus of this project, and is primarily defined so that we can clarify what we mean by "most important features"

II. Data

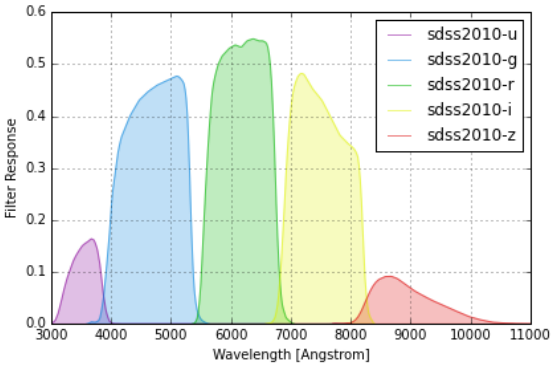
The databases that we will be working with are part of GAMA (Data release 3), the third data release of the Galaxy And Mass Assembly survey.[8]. GAMA is a project that combines ground-based and space-borne survey facilities to study cosmology and galaxy formation and evolution. [8] As a result of this combination, galaxies with a redshift between 0 and 0.3 (see figure 3b) been observed by a variety of different instruments, and at a large amount of different wavelengths and spectral resolutions. This project was chosen as it is both recent, contains a significant number of objects which is a requirement for many machine learning methods, and has already been investigated by several projects, allowing us to compare our findings and methods.

Band	$\lambda_{eff}\ (\mu m)$
u	0.3346
g	0.4670
r	0.6156
i	0.7471
z	0.8918
Z	0.8817
Y	1.0305
J	1.2483
H	1.63133
K	2.2010

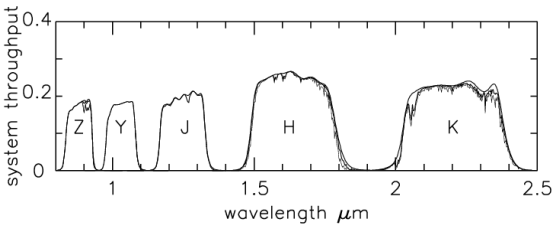
Table 1: *Photometric bands included [7]*

i. Photometric dataset

To make sure we use a consistent dataset we have chosen galaxies (entries) which are only included in the Sloan Digital Sky Survey (SDSS)[9] and in the UKIRT Infrared Deep Sky Survey (UKIDSS)[10]. These entries are not only the most complete, featuring over 90 % of the photometric objects, they are also covered spectroscopically. The wavebands are displayed in figures 2a and 2b plus table 1, where u, g, r, i and z belong to SDSS and Y, J, H and K come from UKIDSS.

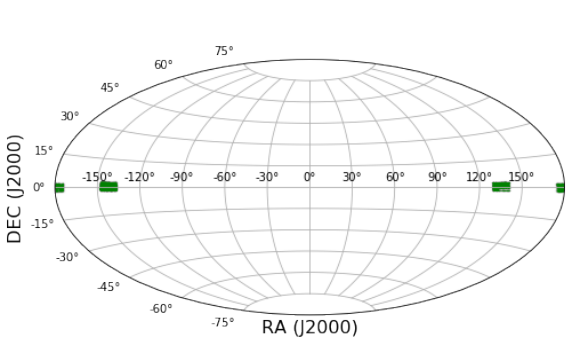


(a) *SDSS transmission curves. [11] [12],*

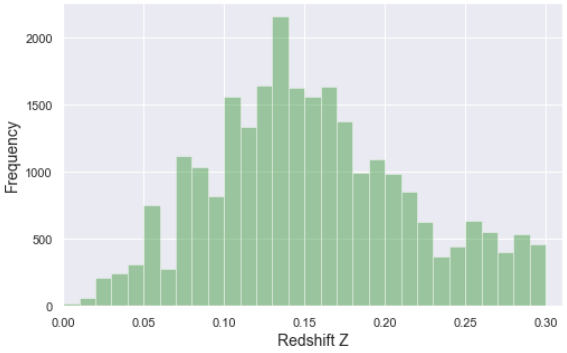


(b) *UKIDSS Transmission curves. [13]*

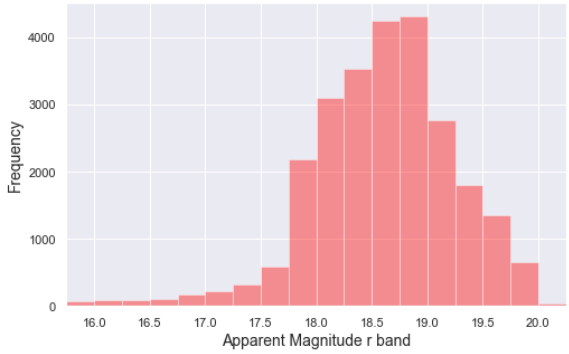
Figure 2: *Overview of of GAMA transmission curves*



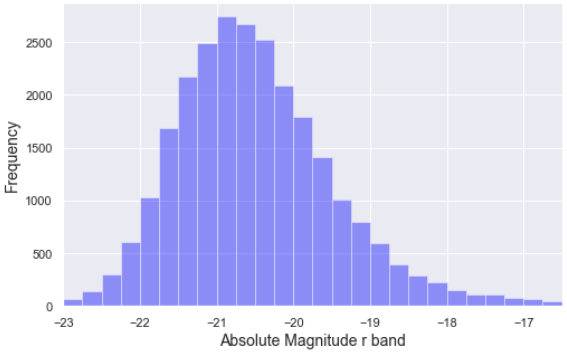
(a) *Aitoff projection of the selected galaxies*



(b) *Histogram displaying redshifts of the selected galaxies*



(c) *Apparent magnitude of our samples*



(d) *Absolute magnitude of our samples*

Figure 3: *Overview of GAMA object galactic locations, redshifts and magnitudes. For the magnitudes, each bin spans 0.25 in magnitude.*

Most photometric information is obtained from the SersicCatSDSS and SersicCatUKIDSS tables. Absolute magnitudes have been determined based on equation 1

$$M_i = 5 + m_i - \left[5 \cdot \log_{10}\left(d.10^6\right)\right] - Kcor_i - Ext_i \tag{1}$$

Where M_i is the absolute magnitude per band, m_i is the initial apparent magnitude per band, d is the distance of the galaxy as determined from redshift, and $Kcor_i$ and Ext_i are the K-correction and relative extinction correction per band. The distance here is based on the redshift z, calculated within the comoving distance frame [14]. For galactic extinction, we have used the galactic extinction catalogue from the GAMA survey [15]. K-Corrections are extracted from the GAMA k-corrections catalogue [16] [17].[3] The sizes have been converted from

[3]following the authors advise we selected the auto fitted magnitudes and used the kcorr_auto_z00.fits table

arcsec to Kpc with trigonometry and the previously calculated distance. Finally, the effective surface brightness remains constant with distance, so no adjusting for the distance required here.

Column name	**Description**	***Units***
CATAID	GAMA catalogue ID number, used as index	
absmag_	Absolute magnitude	*mag*
absmag10re_	Absolute magnitude at 10 times effective radius	*mag*
size90_	Radius of the galaxy that contains 90% of the galaxy light	*kpc*
sizeRE_	Radius of the galaxy that contains 50% of the galactic light (also known as the effective radius)	*kpc*
SersicIndex_	Sersic index of the galaxy	
Ellipticity_	Ellipticity of the galaxy	
MU@0_	Effective surface brightness at centre of the galaxy	*mag / arcsec^2*
MU@E_	Effective surface brightness at the effective radius	*mag / arcsec^2*
MUAVG_	Average effective surface brightness within effective radius	*mag / arcsec^2*

Table 2: *CATAID is constant as an index. All other values repeat per band.*

ii. Spectroscopic dataset

The spectroscopic information is taken from the SpecLineSFR collection included in data release 3 of the GAMA survey[18]. From this collection we use two tables. SpecLineGaussFitSimple contains emission line measurements of 12 different spectral lines derived with single-Gaussian fits, while DirectSummation table provides direct summation measurements for 51 different emission and absorption spectral lines. Both these tables contain contains absorption and emission lines. The 4000 break, or D_{4000} is included in both datasets, and is the ratio between average flux densities on either side of 4000 , and is an indicator on star formation characteristics. Not only do these two datasets represent a different number of lines, but they also differ in how these lines are fitted and how equivalent widths are determined.

Equivalent Width (EW) represents the strength of an absorption or emission line. As explained in figure 4. There are multiple ways to measure such a line from a real instrument. The simplest is perhaps direct summation, where after correcting for redshift, each line detected by a spectrographic pixel array can calculated with equation 2.

$$EW = \Delta\lambda \sum_i \frac{I_{C_i} - I_i}{I_{C_i}}, \qquad (2)$$

where $\Delta\lambda$ is the pixel size, I_{C_i} is the continuum level at the wavelength of the ith pixel, and I_i is the actual flux received by the i'th pixel. [20] The alternative is fitting a Gaussian line to the spectrum, and calculating the area underneath this curve. This has been done by Gordon et al [21]. The method provides an equivalent width (EW) and a flux (F) for the fitted lines, as derived by equation 3:

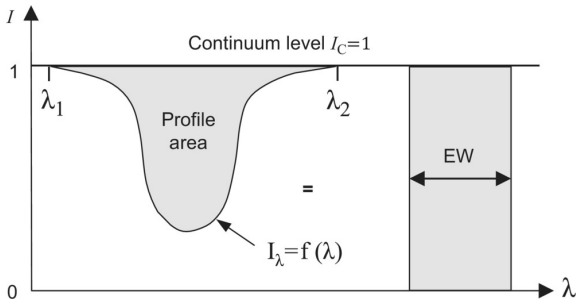


Figure 4: *Equivalent width is based on the profile area of spectral lines. [19]*

$$EW = \frac{F}{C} = \frac{\sqrt{2 * \pi} A \sigma}{C} \qquad (3)$$

Where A is the amplitude of the Gaussian, σ is the line dispersion (including instrument dispersion) and C is the continuum at the position of the emission line given by the linear fit to the continuum. All the equivalent widths are corrected by (1+z) and are thus rest-frame measurements.

We have created datasets where the simple gaussfit and the direct summation are included. In case both systems are used at the same time, recurring Equivalent Widths belonging to the Gaussfit lines are marked by an additional x, while an additional y marks those belonging to direct summation. The spectroscopic dataset for Gaussfit lines contains the EW shown in table 3, while information about the Direc summantion line bands can be found in the appendix.

Name	Wavelength	Comment
OIIB	3726	Tied position, amplitude (0.35*OIIR) and sigma with OIIR, emission only
OIIR	3729	Emission only
Hbeta	4861	Narrow or broad emission and absorption allowed, fitted concurrently with OIIIB/R
OIIIB	4959	Tied position and sigma with OIIIR, emission only
OIIIR	5007	Emission only
OIB	6300	Tied position and sigma with OIR, emission only
OIR	6364	Emission only
NIIB	6548	Tied position and sigma with NIIR, emission only
Halpha	6563	Narrow or broad emission and absorption allowed, fitted concurrently with NIIB/R
NIIR	6583	Emission only
SIIB	6717	Tied position and sigma with SIIR, emission only
SIIR	6731	Emission only

Table 3: *The lines included for the GaussFit spectroscopy subset [18]*

III. Data preparation

We merge the different dataframes within python as pandas dataframes. Using an inner merger on the CATAlogue IDentification number (CATAID), we make sure that different features remain linked to a single CATAID number. Here we also start to filter for erroneous measurements. As all our values are determined beforehand by various fitting tools, we regularly find dummy values for features that could not be adequately fitted. These values occur when there are, for example, insufficient pixels to perform a fit. Possible causes for insufficient pixels can be due to bad pixels or the limits of the wavelength range. The latter can occur when the galaxy is far enough away that the wavelengths of spectral lines have been redshifted beyond the scope of the measurement device.

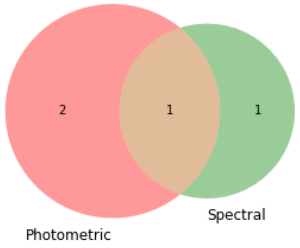


Figure 5: *The center would be the inner merger dataset.*

To thoroughly investigate the feature space of the tables we have selected, we had to decide how to handle these missing values. At first glance, we looked at using either use predictive models or machine learning to make predictions based on trends or physical knowledge and attempt to fill in these blanks. Alternatively, we can fill in these blanks with, for example, the average of that feature. Finally, we can choose to remove any objects for which we do not have a complete set of features. These predictions are beyond the current scope and interest of our project, but more importantly, come with a reasonable likelihood that these predictions might influence any findings. Simply filling in the average is an even worse solution, as this will most definitely influence measures such as variance. As such, we have chosen to discard incomplete objects. In the end, end up with some 25761 galaxies as our complete dataset.

i. Signal to noise and chi squared

For the spectroscopic dataset, a large number of galaxies have had their spectra measured multiple times. By filtering for IS_BEST = True in the simplegauss or DirectSummation tables, we make sure we only select the spectral information that had the best signal to noise ratio and fit results. We also made sure to select only SDSS and UKIDSS survey results and finally we limited ourself to a Signal to Noise Ratio of 3 or higher, as recommended by the creators [21] of these tables.

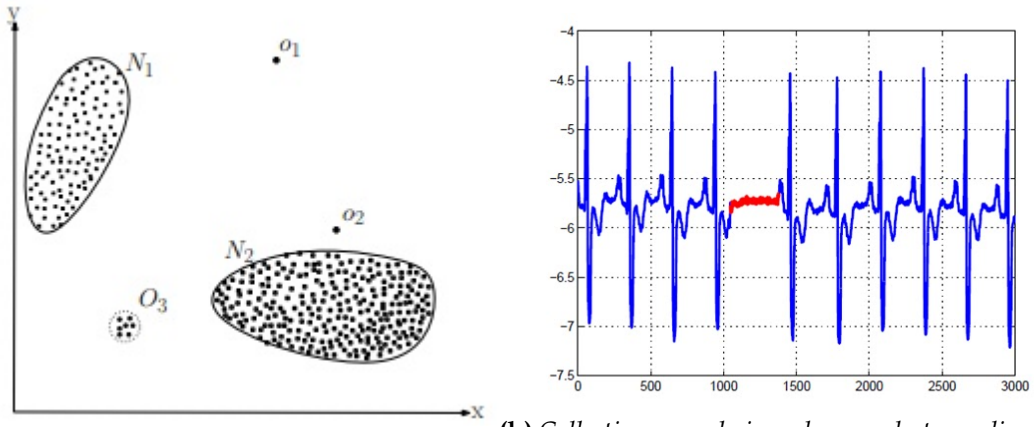
For the photometric dataset, two main estimators of the goodness of fit are provided: GALCHI2 and PRICHI2 in the SersicCat tables. Both are measurements

of the reduced chi-square statistic. GALCHI2 represents the goodness of the GALFIT fit statistics as derived from all pixels in the entire input region, and can also include secondary or background objects. PRICHI2 instead represents the derived goodness of fit statistic based only on those pixels which belong to the primary galaxy source area. As such, PRICHI2 is a more reliable estimator of the model quality. The creators of this catalog (Kelvin et al.[21]) give detailed recommendations, favoring the use of PRICHI2 between the range of > 0.5 and < 1.5, or alternatively > 0.5 and < 2 [22].

Selecting datapoints by filtering on PRICHI2 values between 0.5 and 2 reduces our initial database from 25761 galaxies to 10941 objects, a reduction of more than 57% of our initial dataset. While we have applied all the following methods to this reduced dataset as well (often because smaller datasets are easier to work with when writing code), the need for another method of filtering for erroneous data points seems to be desired.

ii. Anomaly Detection

Anomaly detection (sometimes referred to as outlier detection) are methods of finding patterns in data which do not correspond to the expected behaviour. Outliers are values that deviate significantly from other measurements in a dataset, but quite often the terms anomaly detection and outlier detection are applied interchangeably. These anomalies can be the result of experimental errors, caused by variability in the measurements, or signify a new class of objects. There are uses for anomaly detection in a wide variety of domains, from finding anomalies in MRI imaging or computer network traffic, to credit card transactions or finding anomalies within astronomical data. [23] There are different approaches to anomaly detection available, and the choice often depends upon several aspects. Some of the things to consider are:
The nature of input data. Whether the data we are investigating is univariate or multivariate. Univariate outliers are apparent in the distribution of values within a single feature. Multivariate data can have outliers which are only apparent when multiple features are taken into account at the same time.
The anomaly type: Are we dealing with point anomalies (Anomalies not part of a group, see figure 6a), contextual anomalies (only an anomaly within a particular context, see freezing temperatures in Dutch summer) or collective anomalies (see figure 6b).



(a) *A quick 2 dimensional example. o_1, o_2 and the group O_3 are outliers, while N_1 and N_2 are two distinct groups. [23]*

(b) *Collective anomaly in an human electrocardiogram output. The red values themselves are not anomalous, but when taken as a collective set they are. [23]*

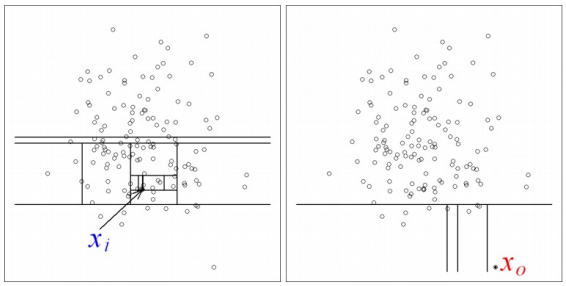
Figure 6: *Examples of anomalies within datasets*

The existence of labels associated with data objects, denoting if an object is normal or anomalous. If these labels exist, they can be used to train a model, and finally the output. Furthermore, whether we are looking to assign labels merely, or if we want to apply a score to each object.
In our case, we are looking to remove errors in measurement or the result of poor fits from our dataset. This means that we are looking for an unsupervised algorithm which can find multivariate point anomalies, and which preferably gives us an anomaly score for each object.

iii. Isolation Forest

The Isolation Forest (IF) anomaly detection method was originally proposed by Liu et al. [24] in 2008. Most approaches to anomaly detection rely upon first constructing a profile of what is "normal", and then identifying instances which do not conform to this normal profile. These methods often have some drawbacks however: they tend to be computationally complex and thus struggle with large datasets that have many features and data entries. And that they are optimized at detecting normal instances rater then actual anomalies, making them likely to detect either too few or too many objects as anomalies (false positives and negatives).

The Isolation Forest method instead takes advantage of two quantitative properties of anomalies: 1: They are the minority, consisting of fewer instances, 2: they have attribute-values that are very different from those of normal instances. In other words, they are 'few and different', which makes them easier to isolate then normal points.

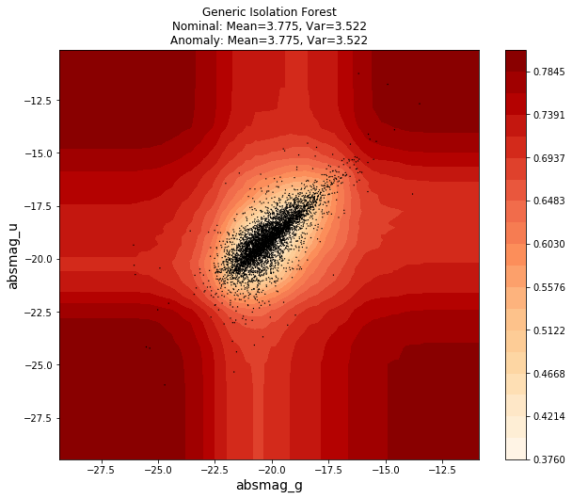


(a) Isolating x_i (b) Isolating x_o

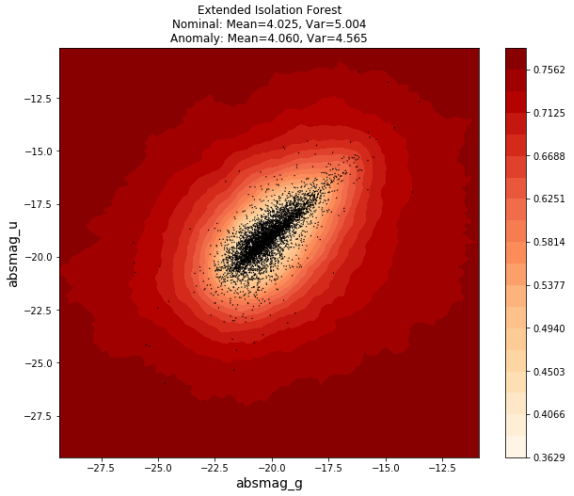
Figure 7: *Example on how isolation forests would be constructed for two different points. [24]*

This method creates many isolation trees around different sub-samples of datapoints randomly pulled from the initial database. The structure of these trees is built by partitioning areas off by comparing samples to randomly chosen feature values. These are the horizontal and vertical lines in figure 7. We repeat this process until a partition contains only a single data point, or in other terms, contains an isolated data point. Anomalies tend to require fewer steps to isolate, so when a forest (collection) of randomly constructed trees consistently produces shorter path lengths for a particular data point then what is normal, we can confidently say we are dealing with an anomalous point. To quantify this, one can derive an anomaly score, which is based on the average path length to a position over the forest of trees compared to the average pathlength of any given tree. A derivation of this can be found in the appendix, and the original paper[24].

The Isolation Forest algorithm is divided into two stages, the training stage, where isolation trees are constructed from a sample ϕ, using the partitioning method described above and an evaluation stage. Here anomaly scores are derived for every object in the dataframe by running them through the ensemble of previously constructed isolation trees. (Following the steps to determine which bin or branch an object belongs in for each tree, then assigning an anomaly score with the formula given above.



(a) *The black dots are datapoints. The contour map represents Isolation Forest anomaly scores. The cross here is an artefact. See text.*



(b) *Contour map of EIF anomaly scores. The artefact is now gone*

Figure 8: *Contour maps of base Isolation Forest (IF) and Extended Isolation Forest (EIF)*

The anomaly scores for a 2-dimensional dataset can be shown in a contour map, for example, as done in figure 8a. Here the anomaly scores for different points in these two feature datasets are displayed. Note here how the contour lines are mostly perpendicular to the axis; however, applying a lower weight to any objects found in these regions. We will return to this artefact momentarily.

Isolation Forest takes a new sample for each tree they construct. This gives them improved performance (less to calculate per tree) and better accuracy, as using a large number of samples can lead to masking and swamping. Swamping is when anomalies are close to ordinary objects, making them harder to detect. Masking is when the existence of too many anomalies within the sample starts to conceal their own presence. With each tree being randomly constructed from a random sampling, this method is not only fast but also accurate.

iii.1 Extended Isolation Forest

The Isolation Forest algorithm does have a flaw; the artefacts we can see in figure 8a. Bias has been introduced by the way the trees are branching. The brighter areas depicted above are a result of all the boundaries of an Isolation Forest being either vertical or horizontal in the relevant feature spaces, as seen in figure 7. These vertical and horizontal divisions create artefacts which can mask anomalies.

There is no fundamental reason though in the concept of IF that restricts us to only horizontal and vertical cuts. The Extended Isolation Forest (EIF) devised by Hariri et al. in 2019 [25] [26] is a variation upon the IF algorithm that uses branch cuts which have random slopes instead. These angled slopes are shown for a 2-D example in figure 9. This change does not significantly alter the speed of the algorithm, yet prevents the formations of artefacts such as seen in figure 8a, resulting in figure 8b.

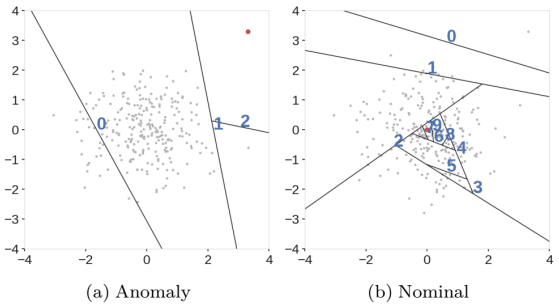


Figure 9: *"Branching process for the Extended Isolation Forest. Only three slices will find the anomaly, while a point near the centre of the clustering requires many cuts"[25]*

For our project, we applied the Extended Isolation Forest algorhitm to our datasets with the settings in table 4. Five hundred (500) trees and 512 samples per tree gave consistent and good results, with more trees not changing output and more samples risking swamping and masking of samples. The EIF paper supports these values .[25] 5% of the objects were dropped based on both good practices within astronomy and after inspection of the anomaly, scores found. With 5%, the maximum score of a dropped point has is less than 0.5. The code can be found in the appendix

Parameter	Value
dropped	5%
ntrees	500
sample_size	512

Table 4: *Parameters by which EIF is applied to the databases*

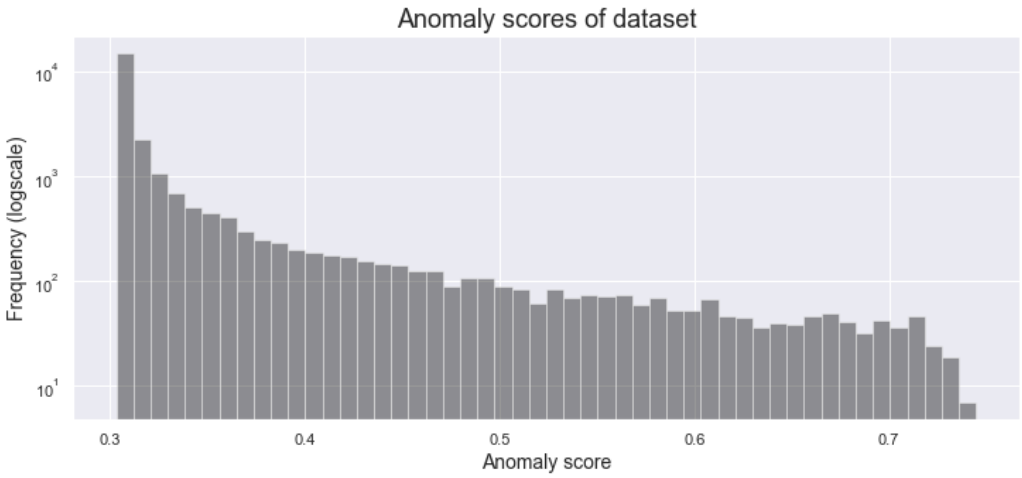


Figure 10: *Histogram of anomaly scores, for the photometric dataset in this case.*

IV. Data Analysis

i. K-means clustering

Before we get started on the actual feature selection, there is one more technique that we need to explain. Both of our feature selection algorithms rely upon this clustering technique, and we will be investigating our final results with this method as well. K-means clustering is a simple and commonly used unsupervised machine learning algorithm. It is meant to group similar data points together and discover underlying patterns. To achieve this objective, K-means looks for a fixed number (k) of clusters in a dataset. With a cluster here, we mean a collection of datapoints aggregated together due to their similarities. Each cluster is defined by a *centroid*, a point at the centre of the cluster. This point can be a datapoint, but more commonly is not. The concept of K-means is to find a k number of centroids, then assign all datapoints to the closest cluster, this, so the centroids remain small. This is an iterative process, which follows the following steps:

```
1 Set k, the number of clusters.
2 Randomly select centroids for these k clusters
3 Assign every datapoint to their closest (euclidean distance)
   centroid
4 Calculate the new centroid for each cluster by taking the mean of
   all datapoints assigned to this cluster
5 Repeat steps 3,4 and 5 until the points converge and the cluster
   centres stop moving.
6
```

[27] [28] Because the initial points are randomly chosen, they can influence the shapes of the final clustering. To mitigate this, we use both k-means++ as a way to select the initial clusters in a smarter fashion and run the k-means algorithm with ten different centroid seeds. This allows us to output the best fit of k-means in terms of inertia from these runs. Inertia here being defined as the sum of squared distances of samples to their closest cluster samples, or in simpler words: we select the result which has the most compact clusters. Harris [28] especially provides excellent visualization of this progress.

Some advantages of K-means clustering over other clustering algorithms: It is a widely used and well understood and studied method. It is quick to train and easy to interpret. It scales well for large datasets, and it will always converge. Some of the disadvantages, however, are: K, the number of clusters, must be pre-determined. K-means is unable to guess by itself how many clusters exist in the data, and other validation methods are required to guess at this. This will be addressed in the results section. It is sensitive to outliers (hence the need for outlier detection), and it assumes that clusters have a spherical shape and to be

evenly sized.

Proceeding forward, we have chosen to use the k-means algorithm primarily because both feature selection algorithms we will be using have been presented with this algorithm in mind. We will, however, evaluate this choice in conclusion, in order to determine if K-means was the correct choice to make.

ii. Types of Feature Selection

- Filter methods select the most relevant features based on intrinsic properties of the data without using any clustering algorithms. These methods are commonly based on statistical or spectral methods
- Wrapper methods use a specific clustering method to evaluate different subsets of features. These methods look to improve the quality of results by applying clustering to these different subsets and then evaluate the results of this clustering to find out which set scores best according to a specific evaluation criterion. The main disadvantage here is that to evaluate all feature subsets comes at a high computational cost, while the method typically can only use a particular clustering algorithm.
- Hybrid methods try to combine qualities of filter and wrapper approaches, trying to create a compromise between efficiency and effectiveness. [6]

iii. Principle Feature Analysis

Principle Feature Analysis, or PFA, is a feature selection method based on the workings of Principle Component Analysis, or PCA. Here, information is inferred from the principal components of the original database in order to obtain an optimal subset of features. While it uses a clustering step, it is still in the category of filter selection algorithms as it clusters only over the collection of eigenvectors. Here we will briefly go over the steps of the actual algorithm. [29] Lu et al. explain PFA in further detail in their paper, and annotated code will be added to the appendix. Starting with A: an $L \times n$ matrix, where L is the number of entries, and n the number of features.
1: We compute the correlation matrix of the dataset. A covariance matrix would have worked as well, as long as we standardized the dataset beforehand.
2: We compute the Eigenvectors and Eigenvalues of the correlation matrix.
3: We pair the eigenvalues with their respective eigenvectors, and sort these in descending order by eigenvalue.
4: We calculate the proportion of variance. By summing up the proportional values

of eigenvalues compared to their total, we can determine how much variance is contained within a number of eigenvectors. 11 Thus if we wish to retain 90% of the variability, we need to select from the sorted eigenvalues until the POV is above 90.
5: With this information, we can construct a matrix A_q from A, using the eigenvectors v_i to v_q, where q is the last eigenvalue required to retain the desired variability. This is pov in our plots
6: We run a clustering algorithm (k-means) on A_q. We look for up to $p >= q$ clusters. p may sometimes be greater than q in case the same retained variability as in PCA is desired. Usually, 1 to 5 features are enough. We call this p_dif in our plots.

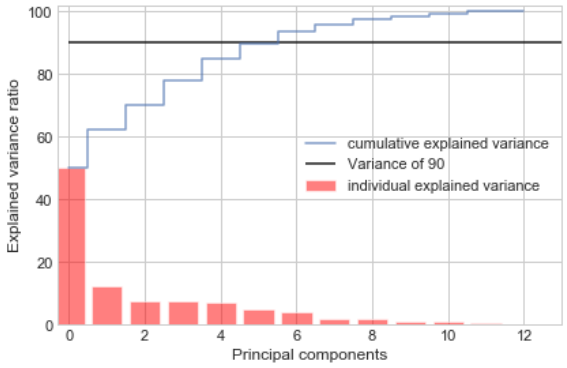


Figure 11: *Proportion of variance. As more eigenvalues are added, more and more of the variance is accounted for*

7: For each cluster, we determine the cluster centre. Then we find whichever vector a_i is closest to this vector, and choose the corresponding feature x_i as a principal feature. This will yield p features, though sometimes the same feature is closest to two centres, and the unique number of features is lower
By choosing the nearest vector, we have found features which have a large spread in the lower dimensional space and are a good representation of the original data. While clustering is used, it is only used in the lower dimensional space ($n \times q$). To find our results, we run PFA multiple times, tallying up our results and for a number of runs, and then seeing which features occur reliably. For clarity sake, any features that were considered relevant in less than 10% of the runs have been removed from the graph, though this number is usually low.

iv. Hybrid method

The hybrid method we have developed by Solorio-Fernández et al. in 2016. [30] It is a filter-wrapper feature selection method that is based on ranking. It was not giving a more exact name, so henceforth when we are talking about the hybrid method, this method is implied. This method consists of two stages: first building a ranking of features with a filter approach (Laplacian Score) and second: selecting relevant feature subsets with a hybrid approach (K-means + WNCH)

iv.1 Laplacian Score (LS)

Laplacian Score is based on a Laplacian matrix. K-nearest neighbour graph, where k is the neighbourhood degree for each instance in the graph. The importance of each feature is evaluated here by its power of locality preservation[31]. It is based on the observation that when two data points are close to each other, they are likely to be related. [32] It is created by constructing a nearest neighbour graph G, deriving a weight matrix S from this and using that a graph Laplacian and Laplacian score. He et al. go into depth on the algorithm and justification, but to grossly oversimplify it: the more influence a feature has on the distance of objects to each other within feature space, the higher the score. A good feature thus should have a small value for the Laplacian matrix belonging to this feature. This way, features can be arranged in a list according to their relevance to the whole dataset. Features at the top of the list will have the smallest values. This way, we can build a list from most to least relevant features regarding data structure preservation. This ordering will then be used to create subsets that will be evaluated in the second stage. Code for Laplacian has been in part been derived from the fsfc package [33].

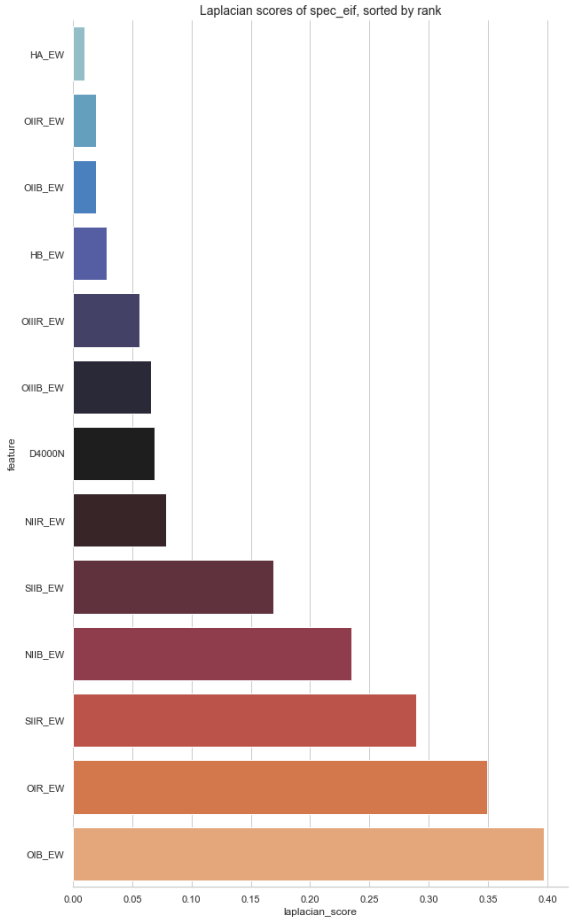


Figure 12: *Laplacian scores for epsilon spectral subset, filtered by rank.*

iv.2 Weighted Normalized Calinski-Harabasz Index (WNCH)

The Calinski-Harabasz index, also known as the Variance Ratio Criterion is a function by which we can evaluate models where we don't have objective information about the labels. This index is the ratio of the sum of between-cluster dispersion and inter-cluster dispersion over all clusters, where dispersion is the sum of distances squared. [34] We have modified this function to be weighted

and normalized as part of the hybrid method. As the value of the CH index tends to increase when features are added, we compensate this by multiplying with the total number of features. And since small Laplacian Scores are better than large ones, we take these into account not only for the ordering but will also divide by the Laplacian score.

$$WNCH(S_o) = \frac{tr(S_b^{X_{S_0}})}{tr(S_w^{(X_{S_0}})} \times \frac{m-c}{c-1} \times |S_0| \times \frac{1}{L_r} \tag{4}$$

Here X_{S_0} is the dataset described by the candidate feature subset S_0, $tr()/tr()$ represents the ration of traces between inter-class and intra-class scatter matrices respectively, m is the number of instances, c is the number of clusters (k for k-means), L_r is the laplacian score associated with the last added or removed feature. $|S_0|$ is the number of features we are calculating WNCH for.

iv.3 LS-WNCH-SR method

Combining the methods described before, we can make a Simple Ranking selection. Keep adding features in order of increasing Laplacian Scores to the feature set, then calculate WNCH score. Find the point where this WNCH score is at its maximum. All features used to reach these points will be selected as most important; any others will be removed.

iv.4 LS-WNCH-BE Method

Modification of before. This Backward Elimination version starts by evaluating the Laplacian score of n features, then removes the least relevant feature. It then applies k-means clustering to a series of these features, dropping the least relevant feature each time and seeing if the WNCH score has improved. If it has, this is considered the new "Best" set of features. After a total of p runs, this process is repeated, once more determining the laplacian scores of the last found "best" subset, then dropping the least relevant features once at a time while clustering and seeing if WNCH improves to a new value of best.

V. FINDINGS

We will apply the previously developed algorithms to two initial datasets, called Beta and Alpha. Both of these datasets contain identical subsets of photometric, spectral and combined features. The only difference is that Beta has been filtered to only contain objects which have PRICHI2 (primary chi-square) values between 0.5 and 2, as detailed in section III.i, now containing 10941 galaxies. Alpha lacks this filter and thus contains the full 25761 objects.

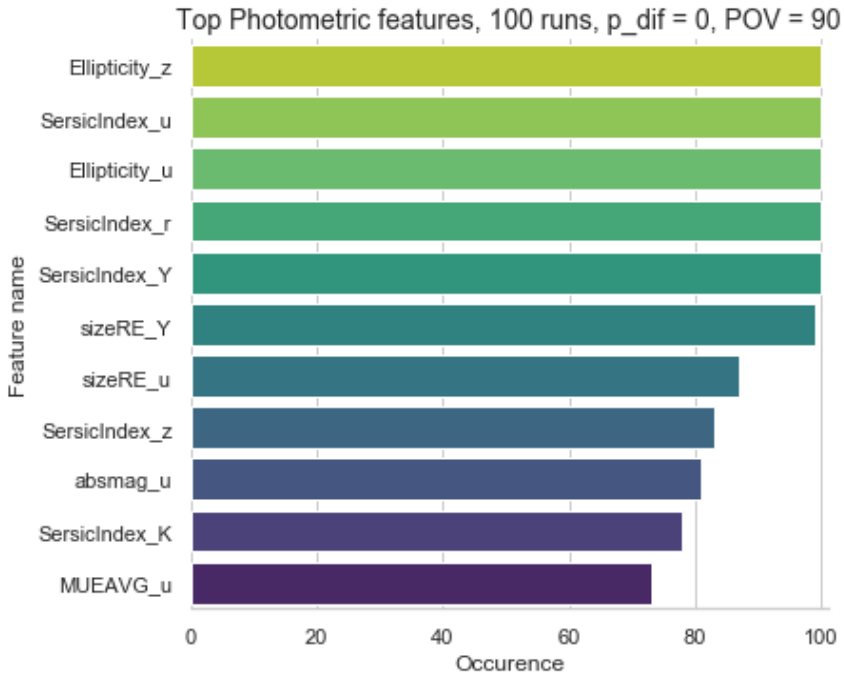
Furthermore, we will be using the terms combined and full dataset in this chapter and the conclusion. Combined implies that it is the photometric dataset plus the Gaussfit subset. Full implies that the dataset contains the photometric dataset and both the Gaussfit and DirectSummation subsets. If only spectroscopic gets mentioned, this implies that we are looking at only the GaussFit, unless otherwise clarified. The first step after selecting a subset of either Alpha or Beta is always to apply the Extended Isolation Forest algorithm. For this, we use 1024 trees and a sample size of 512 per tree to drop 5% of the total number of objects from each subset. After this, we will apply the PFA and Hybrid algorithms to the subsets, including combined sets. We will discuss the results in this section, displaying the most relevant graphs within the text while the other insightful or relevant graphs will be relegated to the appendix or GitHub.

i. Findings PFA

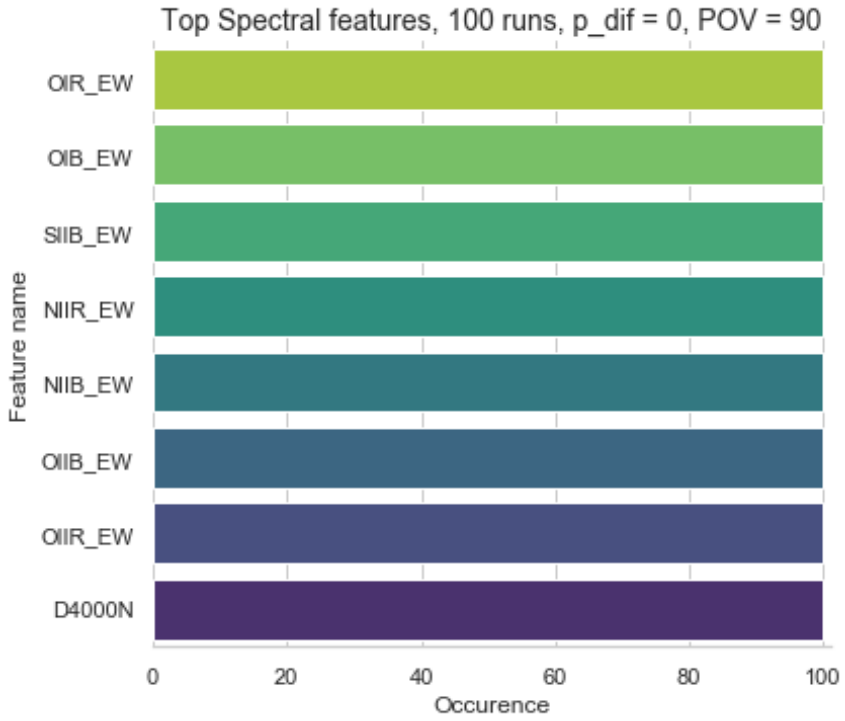
We have made numerous runs with the Principle Feature Analysis algorithm. By changing this algorithm to store results over multiple iterations, we can account for possible variance within the Kmeans step to see if the same features are always considered the most important. Most of these runs have been done with a p differential (p_dif) of 0 and a Proportion Of Variance (POV) of 90, though some runs with different values for these two variables have also been done. These terms have been explained in the section introducing PFA (IV.III.) All these runs have been done 100 times, where we kept track and tallied up the results, allowing a score to be defined based on how often a feature is included as part of the most important features. This r score we define as $r = \frac{\text{\# of runs}}{\text{\# of inclusions}}$, where an r of 1 means this feature is included all the time, and an r of 0.5 means it is only included 50% of the runs.

When we did the same runs for different POV (80, 70 and 95) values the differences were slim or even absent in cases where we lowered the POV, while if we increased the POV for a greater retained portion of the total variance, a sizable

number of extra features were required due to low eigenvalues at this point. When we instead used different values for the p_dif variable (which increases the number of features to be selected in order to conserve variance), the addition of even a single feature lowers both the average r and the r of the topmost features.

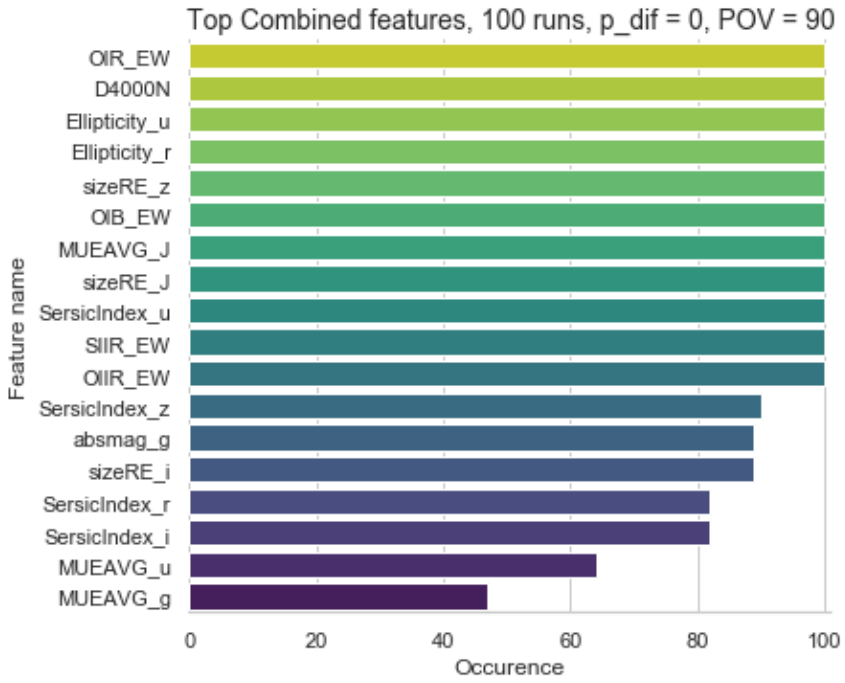


(a) *Photometric features, p_dif = 0*

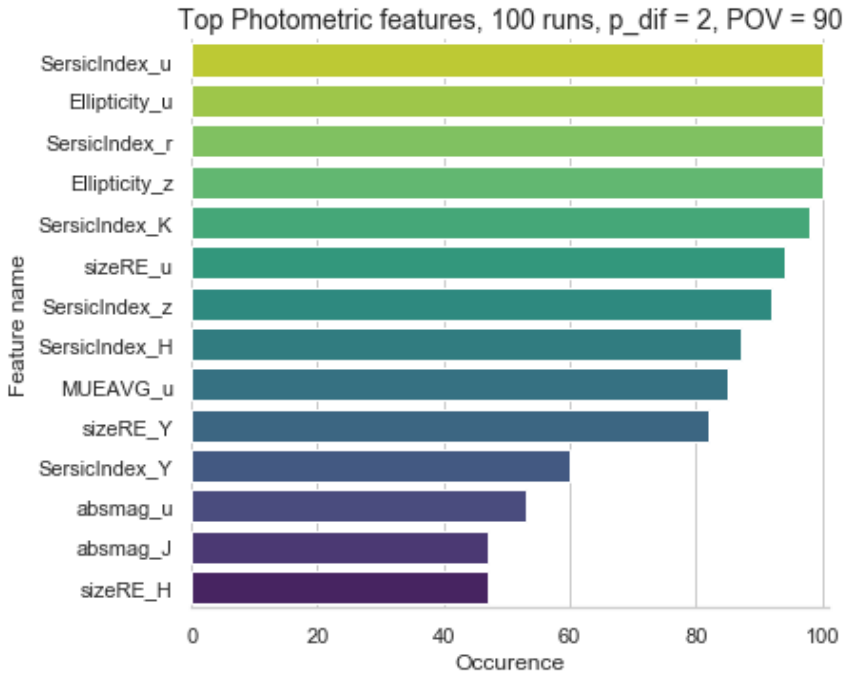


(b) *Spectral features, p_dif = 0*

Figure 13: *Combined results of 100 runs using the PFA algorithm on the Alpha dataset, with POV = 90, p_dif = 0, displaying r > 0.2. (so ignoring any features that occur less then 20% of the time.*

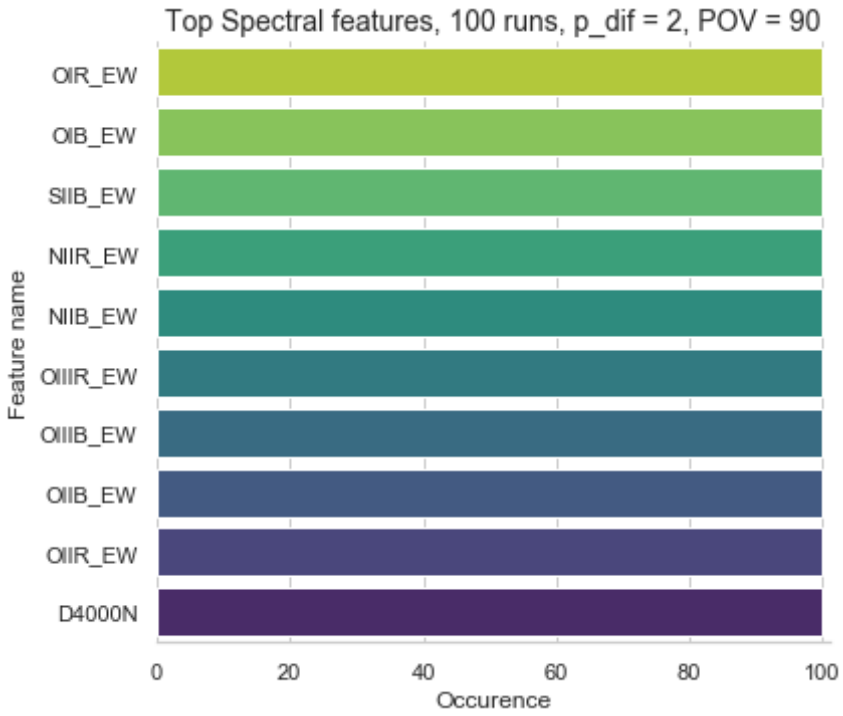


(a) *Combined features, p_dif = 0*

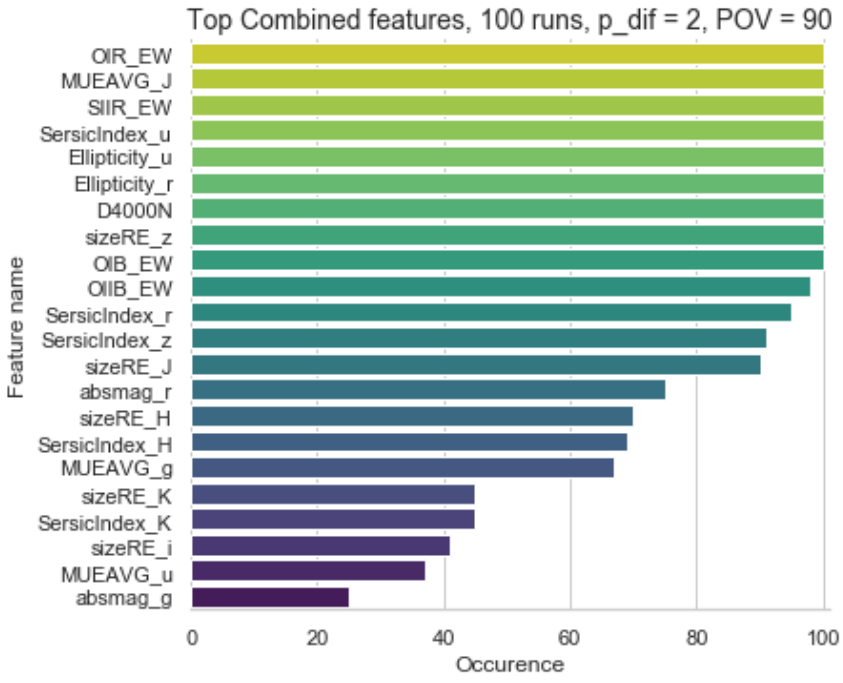


(b) *Photometric features, p_dif = 2*

Figure 14: *Combined results of 100 runs using the PFA algorithm on the Alpha dataset, with POV = 90, p_dif = 0 & 2, displaying r > 0.2. (so ignoring any features that occur less then 20% of the time.*



(a) *Spectral features, p_dif = 2*



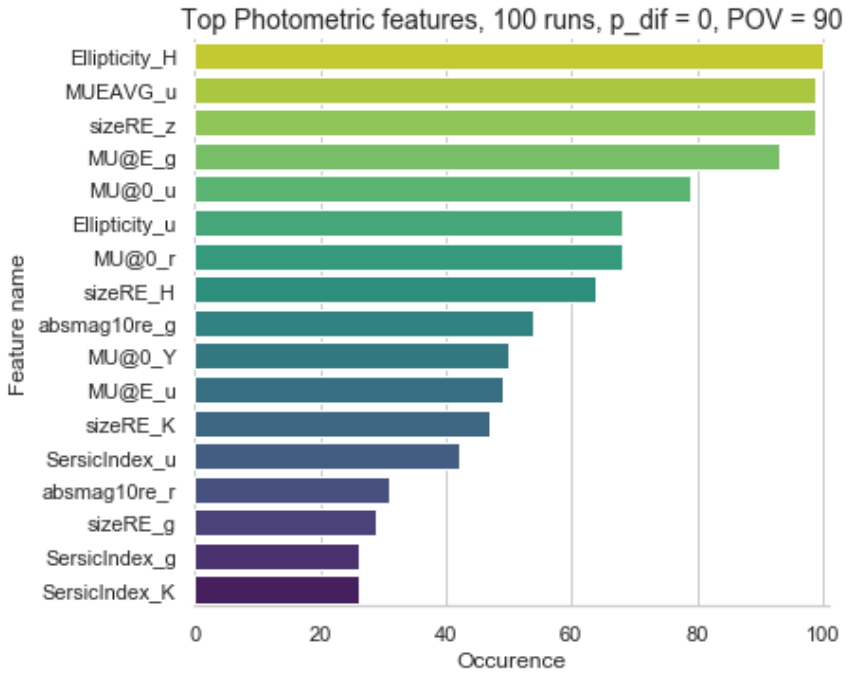
(b) *Combined features, p_dif = 2*

Figure 15: *Combined results of 100 runs using the PFA algorithm on the Alpha dataset, with pov = 90, p_dif = 2, displaying r > 0.2. (so ignoring any features that occur less than 20% of the time.*

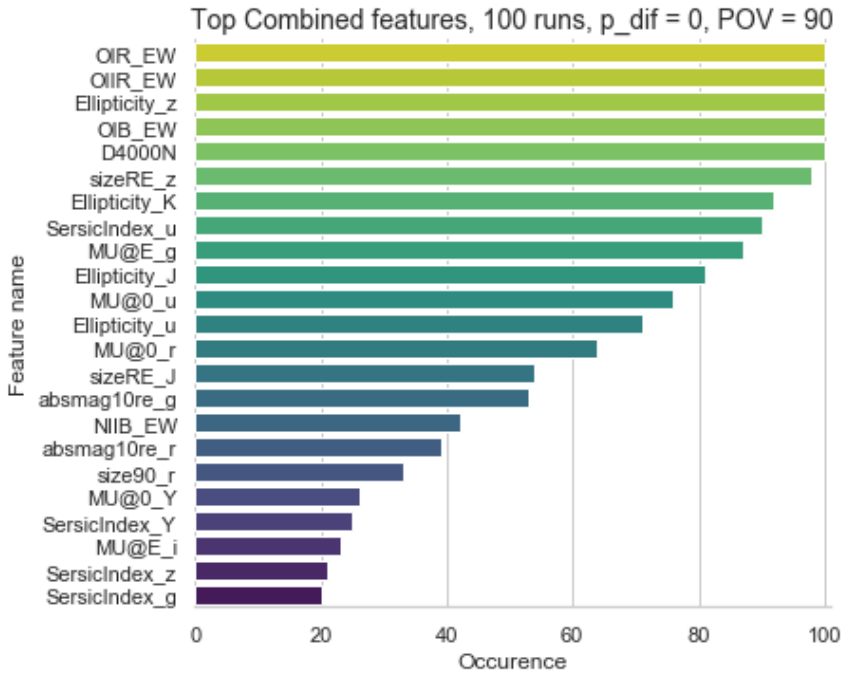
We can see the results in figure 13 14 and 15. For the photometric results, Ellipticity and SersicIndex appear to be of high importance, followed by SizeRE. The most commonly occurring band in our results, however, is u band. For this band, we find the following features of importance: SersicIndex, Ellipticity, SizeRE, absmag and MUEAVG. The z band occurs twice, for Ellipticity and SersicIndex, while the Y band occurs twice as well, for both the SizeRE and SersicIndex features. For (gaussfit) spectral features we have very consistent results, with an r score of 1 (the maximum) for every feature. The features of importance (according to PFA) are thus the equivalent widths of OIR, OIB, SIIB, NIIR, NIIB, OIIB, OIIR and the D4000 break strength. If we set the p_dif at two, two more lines are added, namely OIIIR and OIIIB. Results for the DirectSummation spectral lines can be seen in the appendix. What we can note here is that with a p_dif of two, the average r score is increased. This means that more features were required to conserve all information. The most important lines here are a mix of absorption and emission lines, with only limited overlap compared to the Gausfit spectral features.

We can also compare these findings with the Beta (chi^2 filtered) dataset. The results for this can be seen in figure 16 Here results for photometric and combined features are significantly different compared to the unfiltered dataset. While the spectral lines are almost identical (except for the two added features), both the photometric and combined features quickly decrease in r value, signifying that the algorithm is less effective at finding consistent results with this smaller data sample.

An initial conclusion we can make is that the PFA algorithm gives very consistent results when it comes to the spectral feature set. Not once does it vary from what features it considers the most important. As such, for the criteria of PFA (Selecting a subset of features which conserves the maximum amount of variance), we can confidently say that these are the most important features. For the photometric and combined features, we have a bit more variance. This variance is increased in Beta. By combining the photometric and spectral feature datasets, we can now see that some spectral lines are more important than others when looking at the complete dataset. Given that these spectral lines are also included in our findings for just spectral features, this seems to be consistent. For our datasets, PFA seems to work best with a POV of 90 and a p_dif of 0. Before we draw further conclusions; however, we will first look at the results of the hybrid method.



(a) *Photometric features Beta dataset, p_dif = 0*



(b) *Combined features Beta dataset, p_dif = 0*

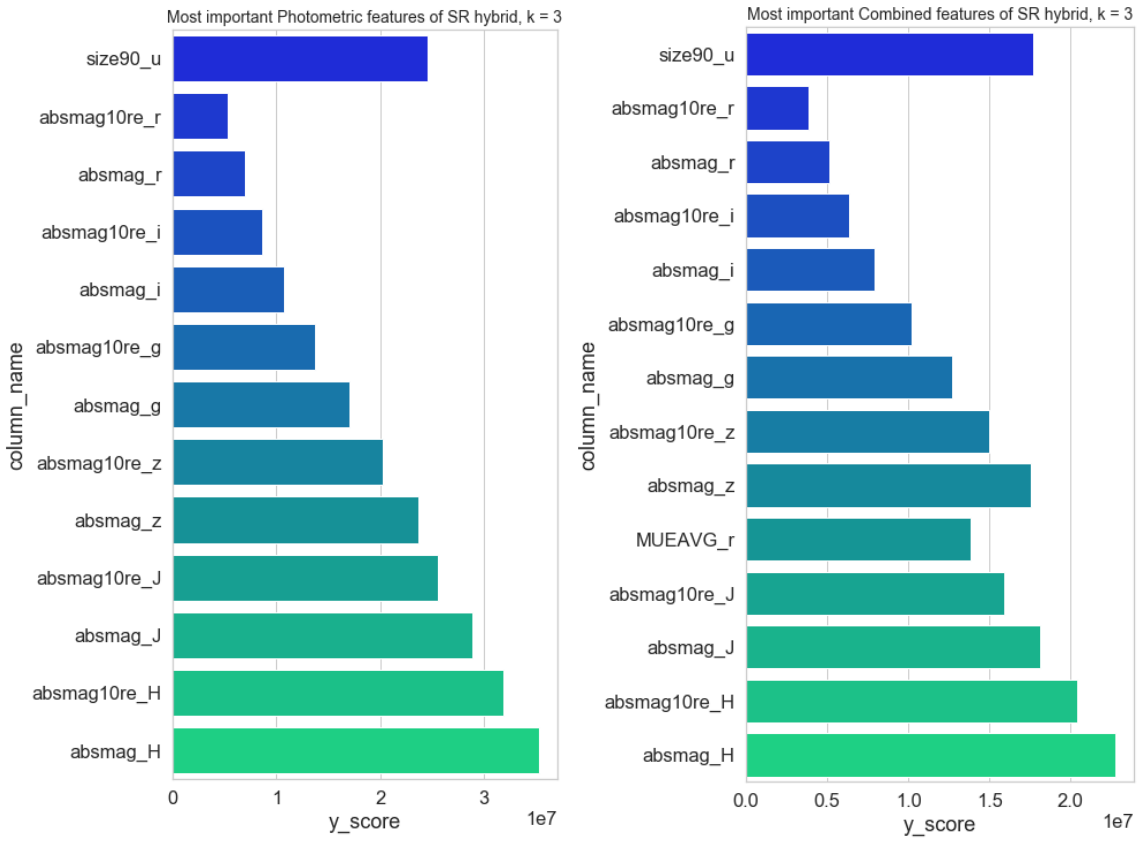
Figure 16: *Combined results of 100 runs using the PFA algorithm on the Alpha dataset, with pov = 90, p_dif = 2, displaying r > 0.2. (so ignoring any features that occur less then 20% of the time.*

ii. Findings Hybrid algorithm

ii.1 LS-WNCH-SR

The Simple Ranking hybrid algorithm (LS-WNCH-SR) has an external variable for the number of clusters (k) to apply. We have run simulations from k = 2 up to k = 10. A consistent result here is that the first variable, size90_u or HA_EW, the size of the galaxy in u band and the Hydrogen Alpha Equivalent width respectively, grows increases faster in value than the other features, soon (or in the case of spectral features instantly) returning only this topmost value as the most important feature to select. We can compensate for this by setting the WNCH value for the first feature to a set number, or by merely plotting the results as a whole and finding the cut-off point. This gives us more features which are of importance. For photometric and combined sets, we can only find non-trivial results for k = 2 and k =3.
The most likely reason for this rapid increase of the top feature, however, seems to be that as the number of clusters increases, it is increasingly challenging to select clear and present clusters. As such, values on a one-dimensional line are easier to separate into roughly equal-sized clusters. We will describe several of the results in-depth, noting any oddities that might be of interest. Further plots can be found in the appendix, on the GitHub, or can be constructed from the code that has been provided in the GitHub.

If we look at figure 17a we can see that size90_u (the radius in which 90 per cent of the galaxies light is contained) is placed as most important by LS rank. In the SR method, the values are ordered by the initial Laplace Score (LS) rankings. There are no other size based features in this set of results, however, and all other results are variants of absmag10re and absmag (absolute magnitude within 10 times the effective radius, and absolute magnitude). The results here are coupled, with absmag10re of a band always slightly ahead of the normal absolute absmag. The bands that occur are in order: r, i, g, z, J and H. For combined, the addition of the spectral lines to this dataset leads to the addition of the MUEAVG_r feature. (Average surface brightness within the effective radius)



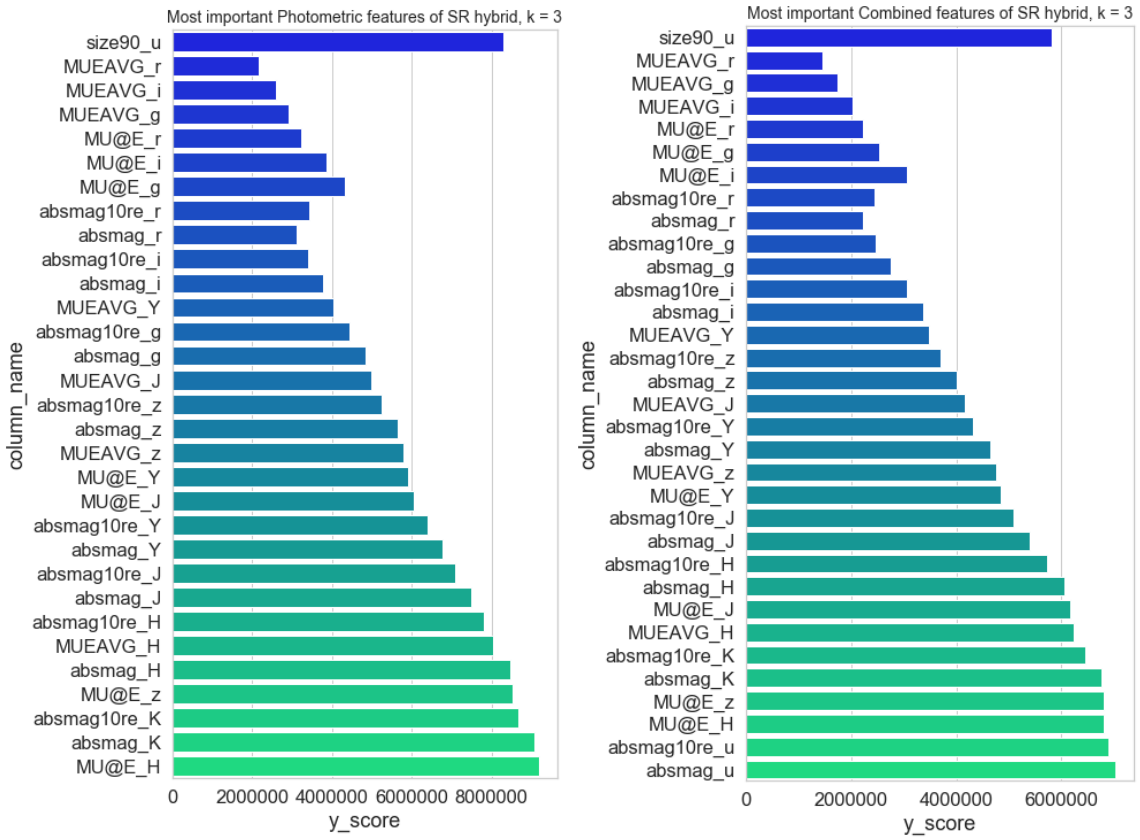
(a) *SR for photometric, k = 3*

(b) *SR for combined, k = 3*

Figure 17: *Results of LS-WNCH-SR for Alpha dataset*

For the beta dataset (figure 18), we find different results. Most importantly, the number of features included has been more than doubled, and now the surface brightness plays a much more dominant role. Again we can see that size90_u is considered the most important feature according to Laplace Score. The next size-related feature (size90_z) is almost halfway down the line in figure 19a, indicating that there is something specific to the size in u band, rather than with all sizes. The next three most important features are all features of MUEAVG (average surface brightness) in r, i and g bands, before being followed by MU@E (surface brightness at effective radius) for the same bands and in the same order. After this, absmag10re and absmag (absolute magnitude at within 10 re and as a whole) of the r band comes up. After this, MUEAVG, MU@E, absmag and absmag10 keep interchanging each other, though frequently the values of the same band are coupled together in order, seemingly indicating some connection. The y_score peaks at MU@E_H, continuing at more or less the same height in value before dropping off sharply after absmag_u. While the size90 of u was in the top, for the

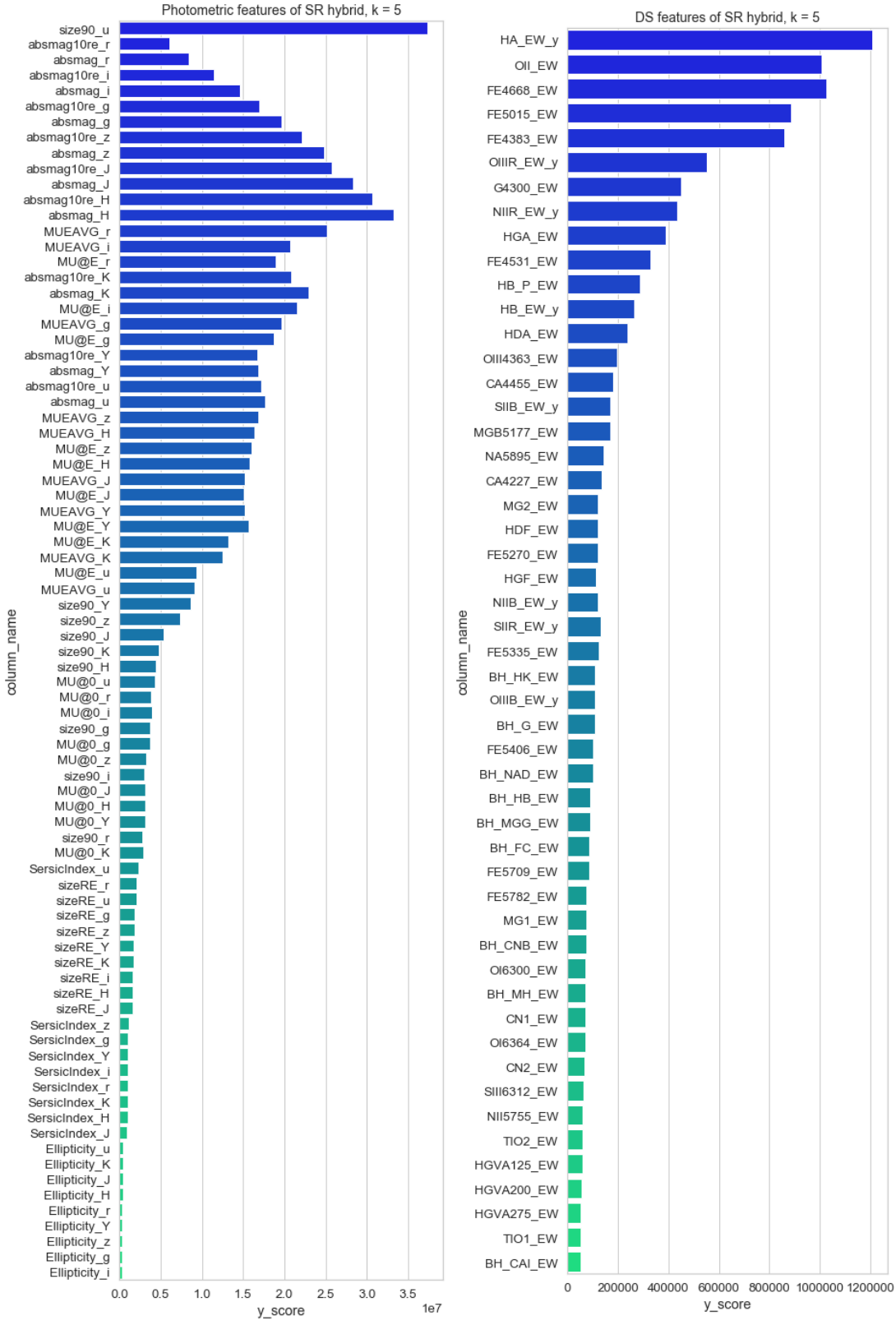
so far mentioned repeating values u band seems to be of lesser importance. The graph 19a also gives some insight into the general importance of the remaining features. Of the lesser important features, MU@0 is tightly clustered and starts shortly after the cut-off point, followed by size90, then sizeRE, then the SersicIndex (with an exception again for the u band) and finally relegating Ellipticity to the least important feature, with the u band being the least important of all Ellipticity features.



(a) *SR for photometric, k = 3*

(b) *SR for combined, k = 3*

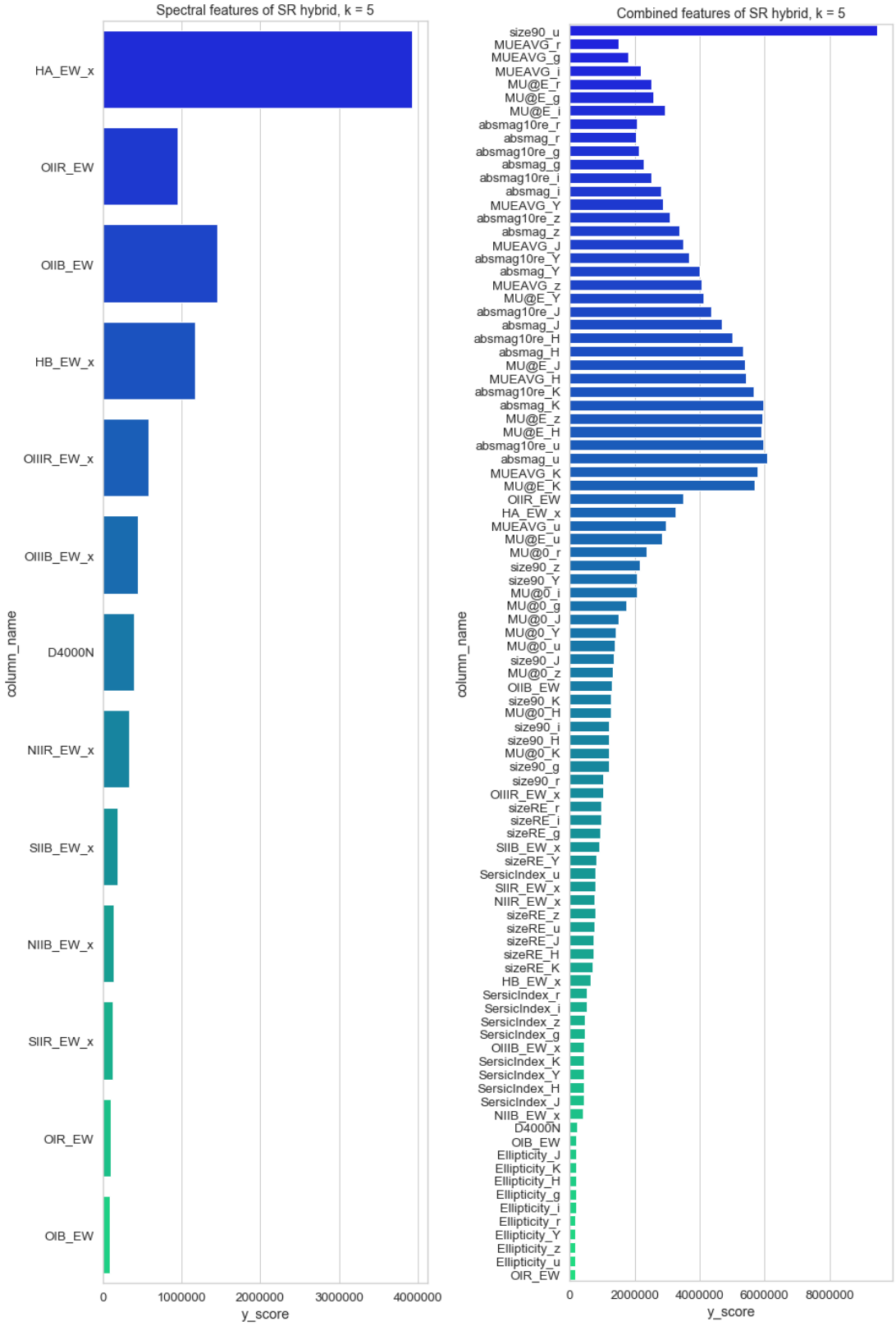
Figure 18: *Results of LS-WNCH-SR for Beta dataset k = 3*



(a) *SR for spectral, k = 5*

(b) *SR for direct summation spectral lines, k = 5*

Figure 19: *Results of LS-WNCH-SR for alpha dataset, k = 5. These are the complete results, not only the most important features, as the first feature dominates otherwise as sole most important feature.*



(a) *SR for spectral, k = 5*

(b) *SR for combined, k = 5*

Figure 20: *Results of LS-WNCH-SR for Beta dataset k = 5. Included the complete results, not just the most important features for k = 5, so that you can see how the first feature increases a lot more quickly then the other features and quickly ends up dominating.*

ii.2 LS-WNCH-BE

The Backwards Elimination version of the hybrid method has an even more substantial amplification effect of the first feature then what we saw with the Simple (Sequential) Ranking. This makes it so that the subset containing only the best Laplacian Score rank (i.e., lowest-scoring number) has a WNCH score higher than any subset containing more features. This applies for any value of k. In order to get greater insights into our results beyond a one feature results, we can modify the WNCH code to either set the result of the primary (first) feature to zero or to divide the result by a set number (in this case 4). These methods have been applied in the results we see in figures 21 and 22. More results can be found in the appendix.

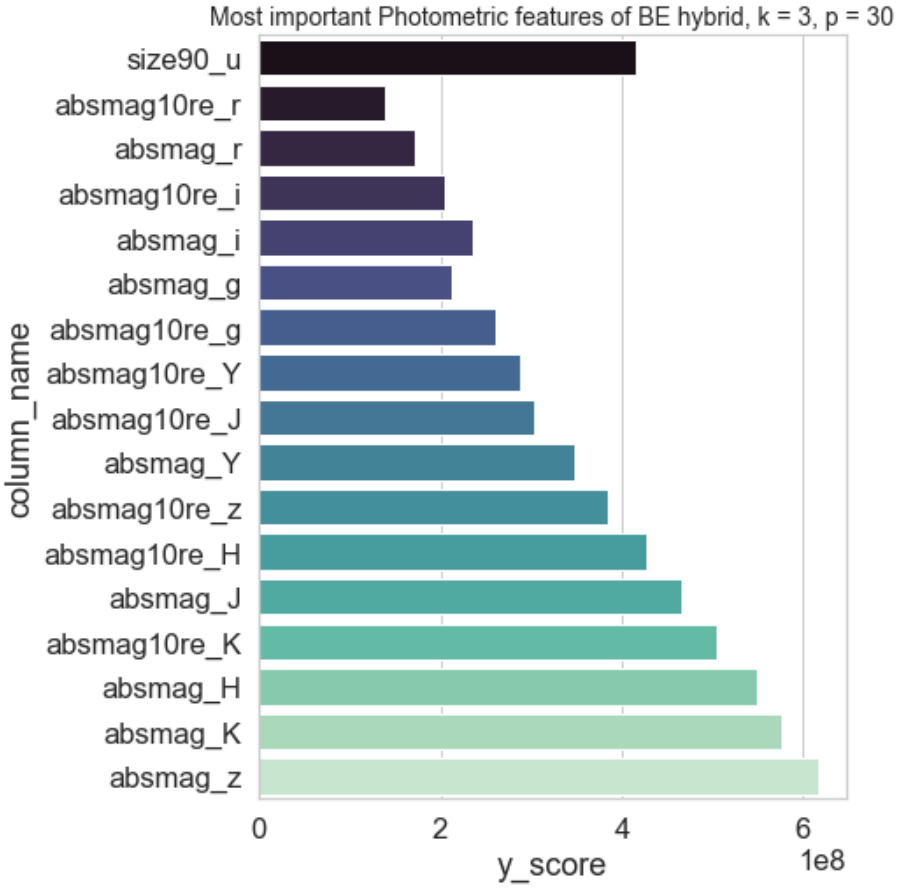


Figure 21: *Most important features according to BE algorithm for Beta photometric dataset, k = 3, p = 30. The primary (topmost) feature has had its value divided by 4.*

Once more, we see that size90_u is on top of the charts. Without dividing

this value by 4, it would be larger than any other feature we examine. The next values are a notable difference compared to the Simple Ranking, as here we can see that all the following features are variants of absmag and absmag10. There is no obvious ordering here, but what stands out is that once more the u band is the odd one out, this time being absent in it is entirety. Now this result does not change even if we add both spectral lines, as figure 22 shows.

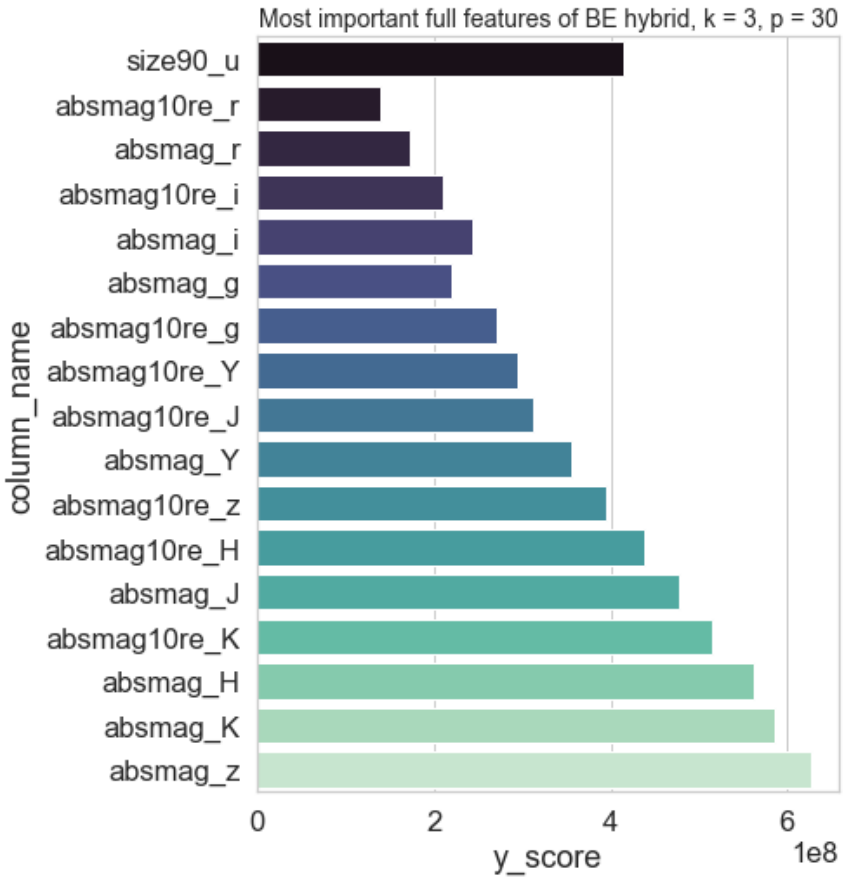
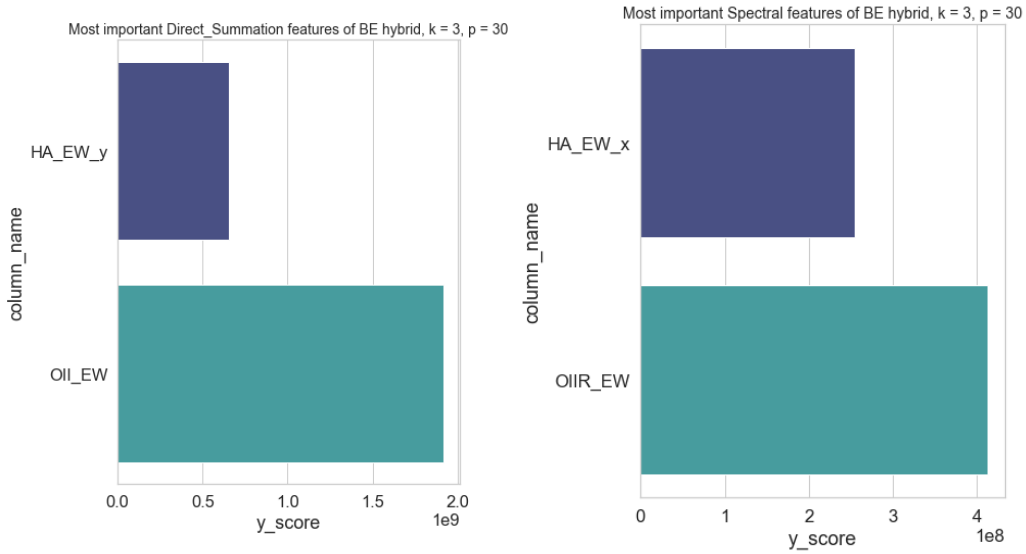


Figure 22: *Results of LS-WNCH-BE for Beta dataset k = 3. The value of the primary feature has been divided by 4. Full means that both the spectral datasets are included in this set.)*

If we look at only spectral lines, we find the results in figure **??**. Here as well, the primary feature has its value quartered. Still, despite the difference in feature space (51 vs 13) features, the results being near as identical. HA, or H-alpha and OII is a spectral line in the Balmer series, which occurs as ionised hydrogen decays from it is n = 3 state to it is n = 2 state. It occurs in regions where hydrogen is being ionised and is used within astronomy as a measurement for star formation rates. A strong (large Equivalent Width) measurement of HA indicates star formation, while older galaxy types will lack this line. As such, it

makes sense that this line is important for classifying galaxies. OII stands for double ionised oxygen, which decays as a doublet at wavelengths of 3 727.092 and 3 729.875 . This doublet is why there is an OIIR and [35] [36]



(a) *Results of BE algorithm upon Beta dataset, Direct Summation spectroscopic lines, k = 3*

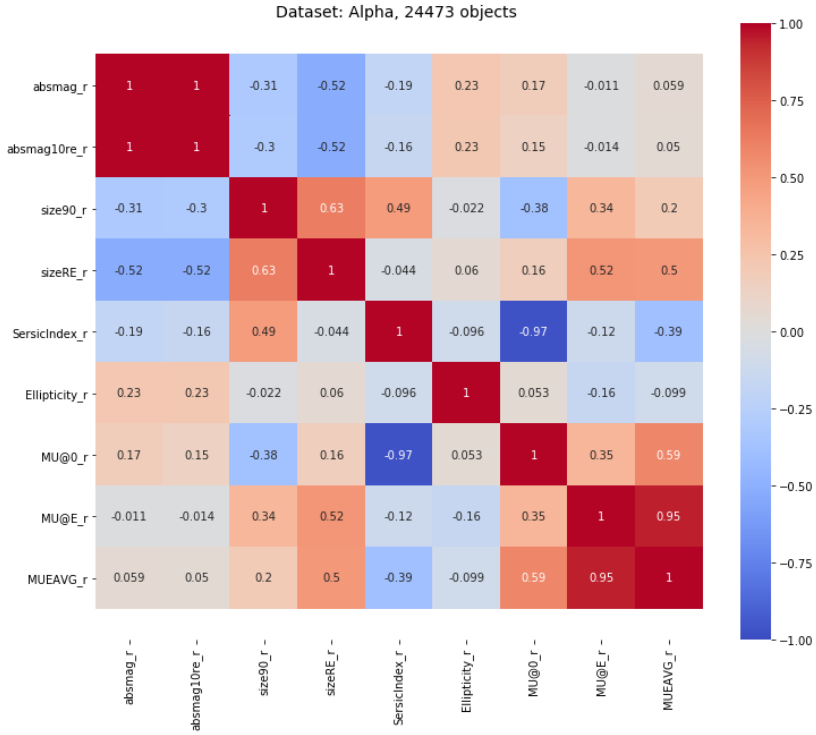
(b) *Results of BE algorithm upon Beta dataset, SimpleGaussFit spectroscopic lines, k = 3*

Figure 23: *Results of LS-WNCH-BE on Beta datasets with k = 3. Primary (First) feature's value is divided by four.*

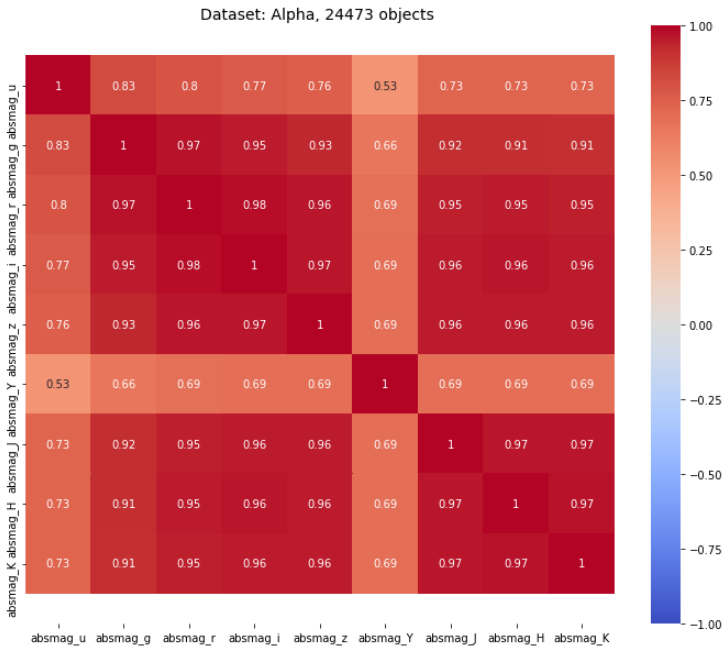
iii. Initial discussion

The results we have found so far are far from perfect. While interesting, a cursory evaluation shows that features of closely of related kinds tend to cluster up in importance. Most of these patterns that we see occurring are in the photometric bands, this either due to some features measuring nearly identical measures (such as absolute magnitude vs absolute magnitude at 10re, or the surface brightness at effective radius versus surface brightness on average) or due to how photometric measurements are repeated over 9 different spectral bands. We shall inspect some correlation matrixes in an attempt to understand this system better, and thereby hopefully propose a way to select an "interesting" initial set of features, or at least one containing less redundancy and irrelevancy. A number of correlation matrix plots can be seen in the appendix figures in figure 24 and in the appendix section of this thesis.

Here again, we find small differences between analysis of the Alpha and the Beta datasets. Some correlations change, but for a few photometric features, we see



(a) *Correlation matrix for r band photometric values*



(b) *Correlation matrix for absolute magnitudes*

Figure 24: *Correlation matrixes for Alpha dataset. Oddities or strong correlations might signify repeated data errors or connections.*

changes in patterns. Namely, if we filter for chi^2 then we find stronger correlations in general for the same photometric feature. For the absolute magnitude we find that the previously weaker correlation in Y band is no longer present, while the correlation of effective radius (sizeRE) is almost double as strong. Finally, the dip in correlation for the i band is no longer present. These changes will be displayed in figure 33.

Some things we can note from these correlations and others we have produced but not included: Similar patterns occur in every waveband. Correlations between different features do not tend to change wildly between bands. There seem to be two exceptions to this, however, u band and in smaller degree Y band. This does, however, depend upon our prior selection. If we filtered our photometric database for a prichi < 2, such as with Beta, then, for example, the Y magnitude is once more correlated akin to other features. Between Spectral and Photometric features, there is only rarely a modest correlation, most of the time the features are uncorrelated. The correlations that we see or the lack thereof can be explained well by our theoretical knowledge. For example, we can expect larger objects to contain more stars and therefore, more starlight. This implies a lower absolute magnitude, and therefore a negative correlation. Objects with a brighter central radius also tend to have a steeper Sersic profile.

Using the correlation matrix, the findings we have obtained so far, and if need be techniques we have developed previously, we can attempt to construct a dataset that gives us more compact findings. First of all, we can determine the number of redundant features. Both the correlation matrix and the pairwise graphs, but especially the hybrid methods imply that the absmag and absmag10re features are redundant. Furthermore, the SR hybrid findings, along with correlation and logical examination, imply that MU@E and MUAVG are mostly redundant.

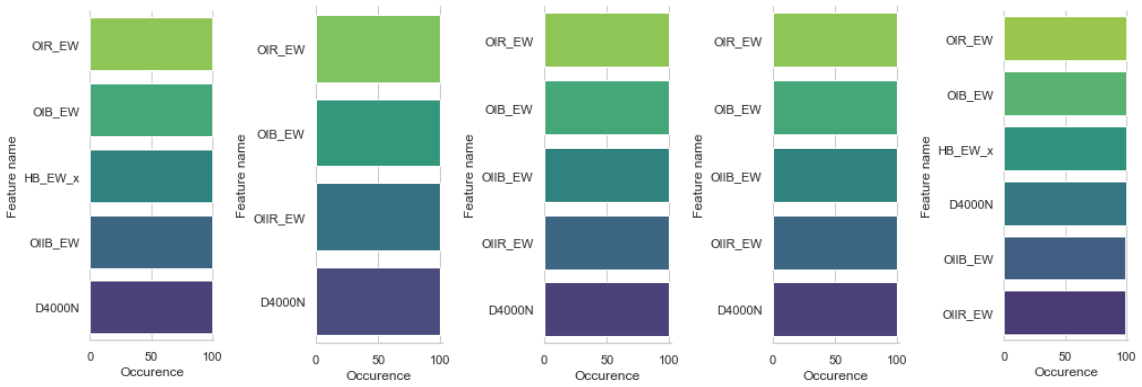
This theory is further supported by applying either Laplacian Scores or PFA to a photometric subset containing only surface brightness values. Here we find MU@0 too be the most important surface brightness feature, followed by MU@E. It seems reasonable to assume that we can remove absmag10re and MUAVG without losing significant information. As such, we can construct a smaller dataset for further feature selection, as well as doing some clustering with primary features we have found.

This removes 2 out of 9 photometric features or 18 features in total. Furthermore, as detailed looks at various datasets have shown, the u band often gives results which do not align with those of other bands. In a smaller fashion, something similar seems to be at play with the Y band. This implies that we might get better results if we remove these two bands entirely.

iv. Gama dataset

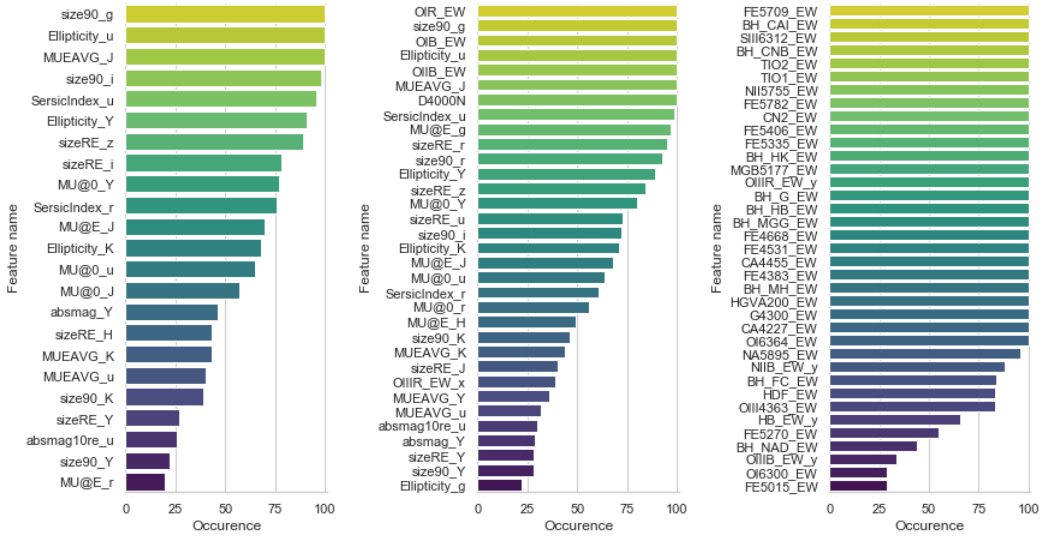
If we use these earlier assumptions to construct a new dataset, we can test the effect of removing these features. After removing the u band, Y band, absmag10re and MUAVG features, we increase the sample size of galaxies (without filtering for chi^2) from 25761 objects to 57594 objects, more than doubling our initial sample. This is mostly due to the fact that a lot of the missing or dummy values that we dropped objects for when we first created our datasets where in these two rows. These steps still include filtering for signal to noise ratios of better than 3. If we do filter for chi^2 however, we still obtain the same 10047 objects that we were working with inside the beta dataset. Currently, computer memory is preventing us from running a full analysis upon a sample size of 57594 objects, as the Laplacian Score step would involve calculating a 57594x57594 matrix, for which significantly more ram is required), but we can make some quick comparisons between the beta dataset and this new dataset that we will call gamma.

In figure 25, we can see some of the results of applying our methods to GAMA. First of all, what stands out is that there are different results for the spectral PFA. The only explanation for how this is possible is that the choice of outliers that are being removed by EIF is critical to the results of PFA. If we instead drop twice as many objects as normal, we get somewhat more consistent results. This behaviour is in part due to by the high sensitivity that PFA has to outliers. That said, there are still consistent results, with OIR, OIB and D4000N being present in every set, and OIIR present in almost any.



(a) *Spectral subset, p_dif = 0* **(b)** *Spectral subset, p_dif = 0* **(c)** *Spectral subset, p_dif = 0* **(d)** *Spectral subset, p_dif = 0* **(e)** *Spectral subset, p_dif = 0*

Figure 25: *Results of PFA on gama spectral datasets, p_dif = 0, POV = 90*



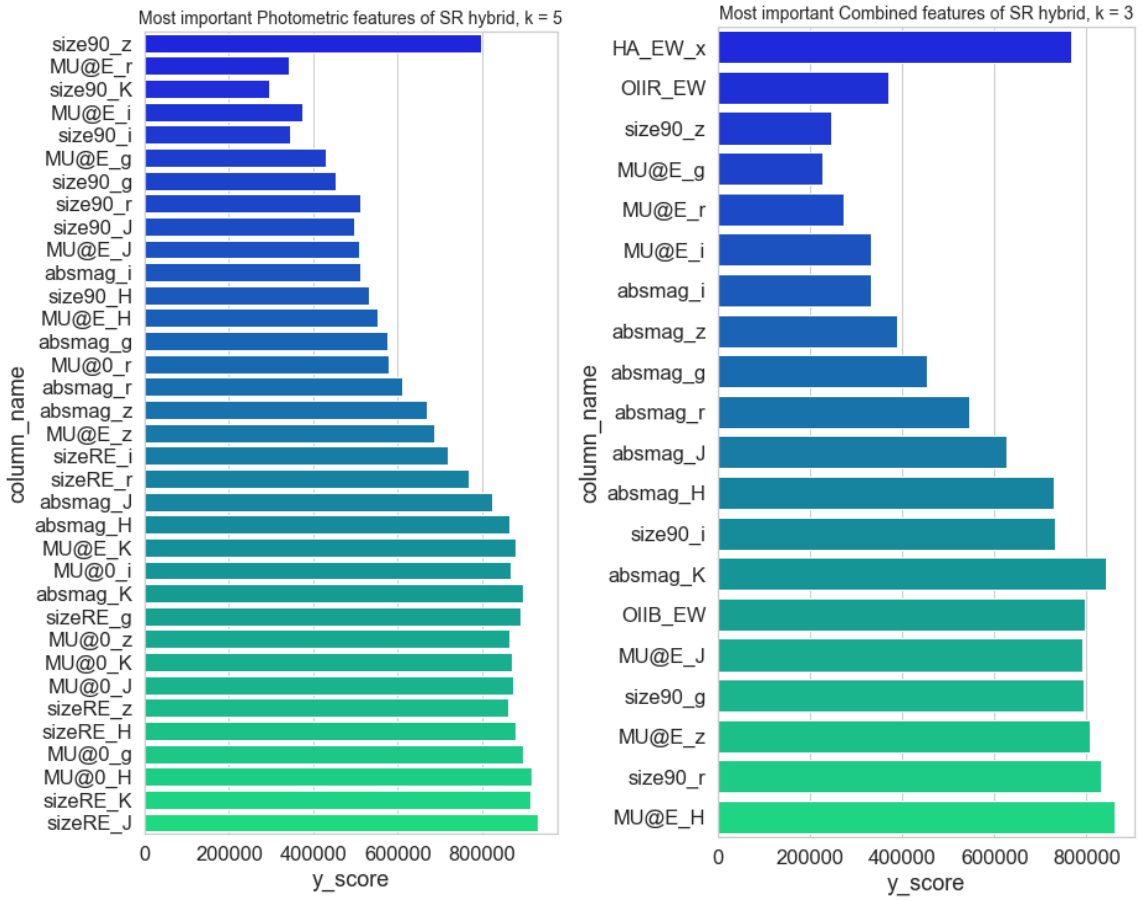
(a) *Photometric features Gama dataset*

(b) *Combined features Gama dataset*

(c) *Direct Summation features Gama dataset*

Figure 26: *Combined results of 100 runs using the PFA algorithm on the GAMA dataset, with POV = 90, p_dif = 0, displaying r > 0.2. (So ignoring any features that occur less then 20% of the time.)*

As we can see in figure 27a the photometric dataset now gives more than a single result for k up to 5. There is still a size90 (size of the galaxy in which 90% of the light fits)on top of the list, but this time size90, along with MU@E (surface brightness at effective radius), take up the 13 most important features, rather than a single isolated band. After this, absolute magnitude plays a more important role, with finally MU@0 (surface brightness at the core of the galaxy) playing a role with 6 different bands as well. All the most important features for clustering with K-means now seem to consist of these four features in various bands. Meanwhile, in figure 27b the combined features at k = 3, we now see HA and OIIR dominate the graph on top. This tells us that these two spectral lines likely indeed have importance.

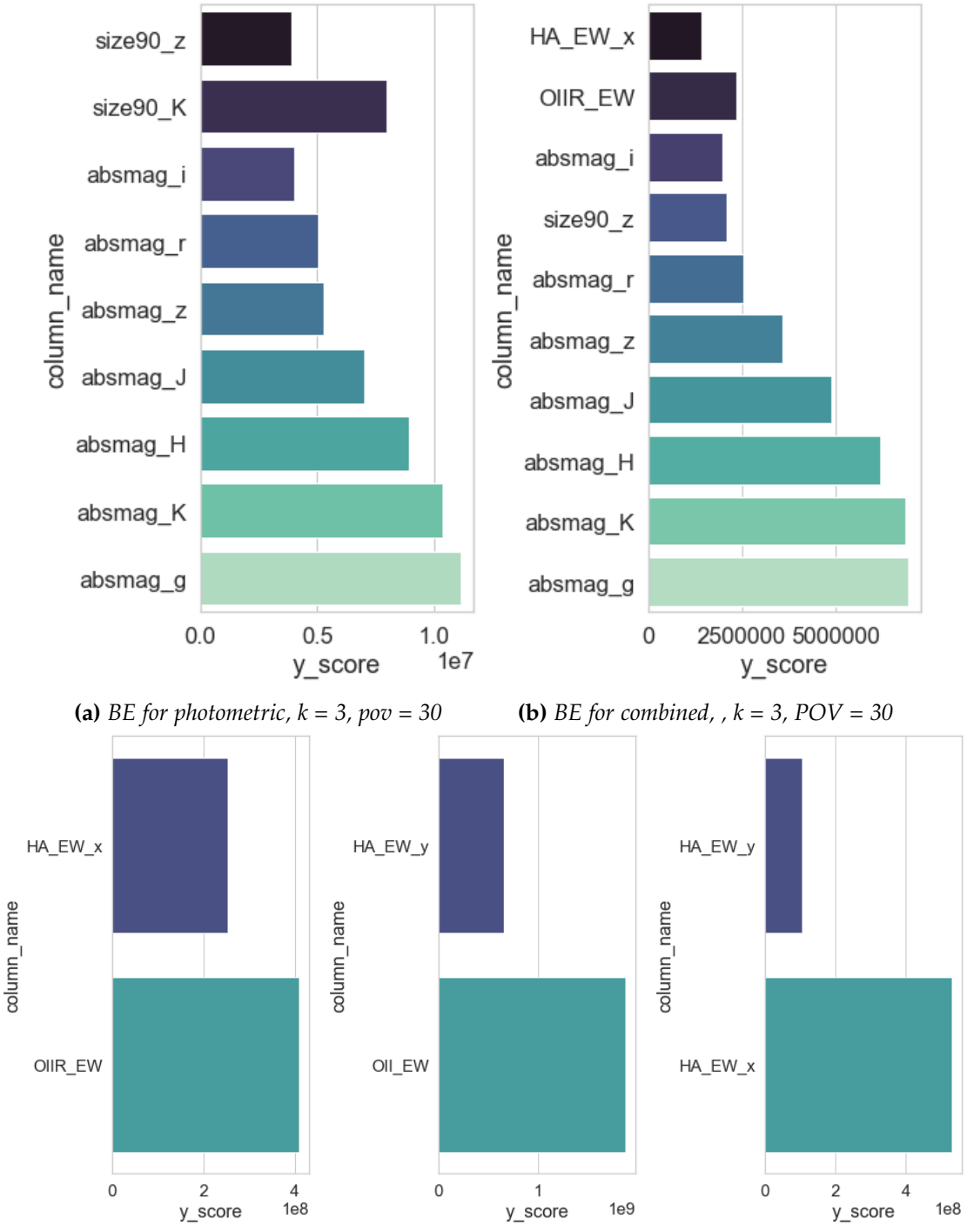


(a) *SR for photometric, k = 5*

(b) *SR for combined, k = 3*

Figure 27: *Results of LS-WNCH-SR for Gama dataset.*

Finally, we have the results of the LS-WNCH-BE algorithm. Here, now that the some of the noise has been removed from u band, we get the findings posted in figure 28. The photometric list has become more clear, with size90 in z and K on top, then the absolute magnitude of the galaxies in all seven bands. The combined results still show HA and OIIR on top, where both belong to the GaussFit spectroscopic subset. size90 for K is now missing, but otherwise, we have the same results as with the photometric set. For the two spectroscopic sets, HA and OII are once more listed as most important. However, what seems to be a very interesting result here is that for the full dataset, so including both spectroscopic subsets and all the photometric features, the only 2 most important features listed is HA for both spectroscopic sets.



(a) *BE for photometric, k = 3, pov = 30*

(b) *BE for combined, , k = 3, POV = 30*

(c) *BE for spectroscopic, k = 3, POV = 30*

(d) *BE for Direct Summation, k = 3, POV = 30*

(e) *BE for full, k = 3, POV = 30*

Figure 28: *Results of LS-WNCH-BE for Gama dataset. k = 3, POV = 30 for all. The primary (first) feature listed has its actual value divided by 4 in these graphs.*

VI. Conclusion

Unsupervised machine learning is a powerful tool, but also a tool that requires thoughtful consideration and aim before implementing it upon a dataset. In order to obtain useful results, one first needs to determine what their goals are exactly. For this project, the goal was to use unsupervised machine learning techniques in order to find the most important features of the GAMA dataset with regards to clustering. Upon reflection, this goal is still too vague and generic, leading us to apply unsupervised machine learning algorithms which are not effective in this case.

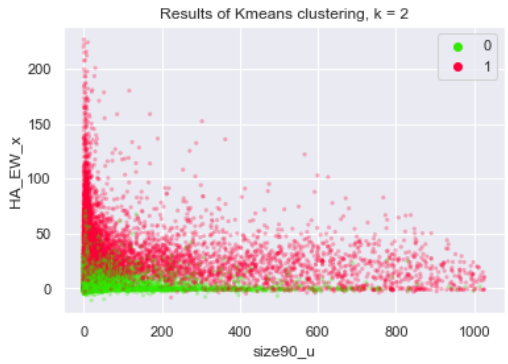
PFA is most suitable for analysing a large amount of very similar features, like sensory data all measuring slightly different points of the same kind of objects, where it excels at determining which of these features contain most of the relevant (non-redundant) information. While Lu [29] has shown that PFA gives good results for its intended application, this does not automatically mean it will improve the results of clustering algorithms. The hybrid method, on the other hand, is aimed to find the best features based on clustering methods.

This difference in approach or internal goals causes the PFA and Hybrid methods to give conflicting results. The most striking example of this is the change in relevance for Ellipticity and SersicIndex. This is due to these algorithms having different goals in mind, and as a result of these different placing levels of importance upon variance.

For the Hybrid method, the initial Filter step is based on Laplacian Scoring (LS). For a feature to have a good (low) Laplacian Score, the feature should have little effect on the relative distances between objects in the dataset. It is meant to select features which conserve locality [31], resulting in a subset where objects are kept close together. Variance plays no direct role here, and we might even logically assume, though no proof will be given, that when features have a high amount of variance, they will have a greater effect on distances between objects. Principle Feature Analysis (PFA), as explained in chapter iv.iii, is based upon PCA, and is aimed at finding the smallest possible subset of features to conserve information about the dataset as a whole. This is done by projecting data along with the directions of maximal variance. Here, features with more variance are of greater importance.

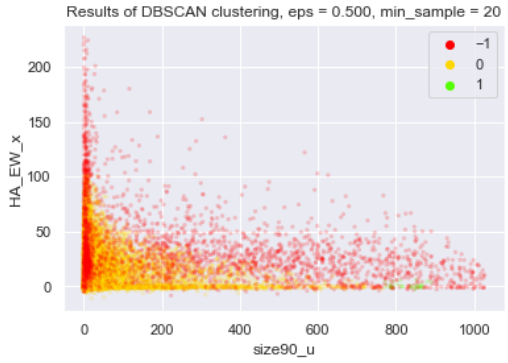
While Solorio-Fernandez [30] proves algorithm to be efficient within his the original paper, we have so far only applied it with the k-means clustering algo-

rithm. This algorithm was chosen because it was used in the original hybrid and thus well tested, plus it is a well understood and efficient clustering algorithm. K-means has the disadvantage of requiring K to be set by the user, however, and in expecting clusters to be of equal size. Instead of using centroid-based clustering like K-means, density-based clustering might be an effective alternative within the field of astronomy. These work by identifying dense clusters of points, allowing them to form clusters of arbitrary shapes and identify outliers. An example of this is the DBSCAN algorithm [37]. This algorithm determines for each point if it's an outlier or part of a greater cluster, the applies labels for each distinct cluster. While we will not go into further detail of these methods here, we will display results of these algorithms a set of features selected for their apparent importance over multiple results.: [['size90_u','SersicIndex_r','HA_EW_x','OIIB_EW','u-i','z-J']]. Here u-i and z-J make up the relative magnitudes or colour of objects, as determined by absolute magnitudes.

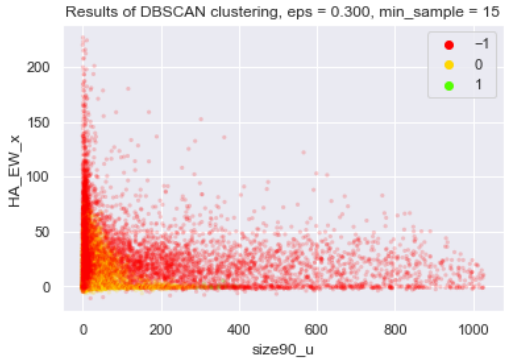


(a) *K-means clustering on k = 2*

(b) *K-means clustering on k = 3*



(c) *Red points are outliers. Note the small green cluster bottom right*



(d) *Red points are outliers.*

Figure 29: *Scatter plots for clustering results*

Not all clustering methods work well on large datasets, and many clustering methods struggle as the number of features increases. For this, unsupervised feature selection can be beneficial. However, for unsupervised feature selection to be truly effective, we need to take a more in-depth look at the various clustering methods available or determine our ways to best filter out redundant and irrelevant objects.

We hope that the Gama dataset, however, has shown the potential and possible strength for these algorithms. By applying the used algorithms to either more suitable problems within the field of astronomy or by using another clustering method as mentioned before, these techniques will have their chance to shine. Examples for more suitable problems could be for PFA: the selection of principle features within image detection systems. We believe that with the right clustering method and tweaks, the hybrid method can produce more unambiguous results with regards to the classification of galaxies.

On a final positive note, the Extended Isolation Matrix seems to perform well at applying outlier detection to astronomical databases. This algorithm can be applied efficiently to many different datasets. We will supply python code for EIF, PFA and Hybrid method on the Github [4] so that others can use, modify and implement these for their future projects, thereby hopefully making this project a useful starting point for further research.

[4] `https://github.com/Dainean/astro_UFS`

References

[1] G. Dr and G. Helpdesk, "Everything you wish you had known before you started working with Gaia Data Release 2," no. 1, p. 21, 2020.

[2] M. Heller, "What is machine learning? Intelligence derived from data," May 2019, library Catalog: www.infoworld.com. [Online]. Available: https://www.infoworld.com/article/3214424/what-is-machine-learning-intelligence-derived-from-data.html

[3] A. Zimek, E. Schubert, and H.-P. Kriegel, "A survey on unsupervised outlier detection in high-dimensional numerical data," *Statistical Analysis and Data Mining*, vol. 5, no. 5, pp. 363–387, Oct. 2012. [Online]. Available: http://doi.wiley.com/10.1002/sam.11161

[4] Z. A. Zhao, H. Liu, and H. Liu, *Spectral Feature Selection for Data Mining*. Chapman and Hall/CRC, Dec. 2011. [Online]. Available: https://www.taylorfrancis.com/books/9780429107191

[5] J. G. Dy, "Feature Selection for Unsupervised Learning," p. 45.

[6] S. Solorio-Fernández, J. A. Carrasco-Ochoa, and J. F. Martínez-Trinidad, "A review of unsupervised feature selection methods," *Artificial Intelligence Review*, vol. 53, no. 2, pp. 907–948, Feb. 2020. [Online]. Available: https://doi.org/10.1007/s10462-019-09682-y

[7] "UKIDSS Technical Page." [Online]. Available: http://www.ukidss.org/technical/photom/photom.html

[8] "GAMA | Galaxy And Mass Assembly." [Online]. Available: http://www.gama-survey.org/

[9] "SDSS." [Online]. Available: https://www.sdss.org/

[10] A. Lawrence, S. J. Warren, O. Almaini, A. C. Edge, N. C. Hambly, R. F. Jameson, P. Lucas, M. Casali, A. Adamson, S. Dye, J. P. Emerson, S. Foucaud, P. Hewett, P. Hirst, S. T. Hodgkin, M. J. Irwin, N. Lodieu, R. G. McMahon, C. Simpson, I. Smail, D. Mortlock, and M. Folger, "The UKIRT Infrared Deep Sky Survey (UKIDSS)," *Monthly Notices of the Royal Astronomical Society*, vol. 379, no. 4, pp. 1599–1617, Aug. 2007, arXiv: astro-ph/0604426. [Online]. Available: http://arxiv.org/abs/astro-ph/0604426

[11] M. Doi, M. Tanaka, M. Fukugita, J. E. Gunn, N. Yasuda, e. Ivezić, J. Brinkmann, E. de Haars, S. J. Kleinman, J. Krzesinski, and R. F. Leger, "PHOTOMETRIC RESPONSE FUNCTIONS OF THE SLOAN DIGITAL SKY SURVEY IMAGER," *The Astronomical Journal*, vol. 139, no. 4, pp. 1628–1648, Apr. 2010. [Online]. Available: https://iopscience.iop.org/article/10.1088/0004-6256/139/4/1628

[12] Speclite, "Filter Response Curves." [Online]. Available: https://speclite.readthedocs.io/en/latest/filters.html

[13] P. C. Hewett, S. J. Warren, S. K. Leggett, and S. T. Hodgkin, "The UKIRT Infrared Deep Sky Survey ZY JHK photometric system: passbands and synthetic colours," *Monthly Notices of the Royal Astronomical Society*, vol. 367, no. 2, pp. 454–468, Apr. 2006. [Online]. Available: https://academic.oup.com/mnras/article-lookup/doi/10.1111/j.1365-2966.2005.09969.x

[14] "Table — Astropy v4.2.dev147+g499195018." [Online]. Available: https://docs.astropy.org/en/latest/api/astropy.table.Table.html#astropy.table.Table

[15] "GAMA | Galaxy And Mass Assembly." [Online]. Available: http://www.gama-survey.org/dr3/schema/table.php?name=GalacticExtinction

[16] "GAMA | Galaxy And Mass Assembly." [Online]. Available: http://www.gama-survey.org/dr3/schema/dmu.php?id=7

[17] J. Loveday, P. Norberg, I. K. Baldry, S. P. Driver, A. M. Hopkins, J. A. Peacock, S. P. Bamford, J. Liske, J. Bland-Hawthorn, S. Brough, M. J. I. Brown, E. Cameron, C. J. Conselice, S. M. Croom, C. S. Frenk, M. Gunawardhana, D. T. Hill, D. H. Jones, L. S. Kelvin, K. Kuijken, R. C. Nichol, H. R. Parkinson, S. Phillipps, K. A. Pimbblet, C. C. Popescu, M. Prescott, A. S. G. Robotham, R. G. Sharp, W. J. Sutherland, E. N. Taylor, D. Thomas, R. J. Tuffs, E. van Kampen, and D. Wijesinghe, "Galaxy and Mass Assembly (GAMA): ugriz galaxy luminosity functions," *Monthly Notices of the Royal Astronomical Society*, vol. 420, pp. 1239–1262, Feb. 2012. [Online]. Available: http://adsabs.harvard.edu/abs/2012MNRAS.420.1239L

[18] "GAMA | Galaxy And Mass Assembly." [Online]. Available: http://www.gama-survey.org/dr3/schema/dmu.php?id=8

[19] M. F. M. Trypsteen and R. Walker, *Analysis of the Spectra*. Cambridge University Press, 2017, p. 76–84.

[20] P. B. Stetson and E. Pancino, "DAOSPEC: an automatic code for measuring equivalent widths in high-resolution stellar spectra," *Publications of the Astronomical Society of the Pacific*, vol. 120, no. 874, pp. 1332–1354, Dec. 2008, arXiv: 0811.2932. [Online]. Available: http://arxiv.org/abs/0811.2932

[21] Y. A. Gordon, M. S. Owers, K. A. Pimbblet, S. M. Croom, M. Alpaslan, I. K. Baldry, S. Brough, M. J. I. Brown, M. E. Cluver, C. J. Conselice, L. J. M. Davies, B. W. Holwerda, A. M. Hopkins, M. L. P. Gunawardhana, J. Loveday, E. N. Taylor, and L. Wang, "Galaxy and Mass Assembly (GAMA): active galactic nuclei in pairs of galaxies," *Monthly Notices of the Royal Astronomical Society*, vol. 465, pp. 2671–2686, Mar. 2017. [Online]. Available: http://adsabs.harvard.edu/abs/2017MNRAS.465.2671G

[22] "GAMA | Galaxy And Mass Assembly." [Online]. Available: http://www.gama-survey.org/dr3/schema/dmu.php?id=11

[23] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly Detection: A Survey," *ACM Comput. Surv.*, vol. 41, Jul. 2009.

[24] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation Forest," in *2008 Eighth IEEE International Conference on Data Mining*. Pisa, Italy: IEEE, Dec. 2008, pp. 413–422. [Online]. Available: http://ieeexplore.ieee.org/document/4781136/

[25] S. Hariri, M. Carrasco Kind, and R. J. Brunner, "Extended Isolation Forest," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2019, conference Name: IEEE Transactions on Knowledge and Data Engineering.

[26] S. Hariri, "sahandha/eif," May 2020, original-date: 2018-07-02T16:49:30Z. [Online]. Available: https://github.com/sahandha/eif

[27] "K-means Clustering Python Example | by Cory Maklin | Towards Data Science." [Online]. Available: https://towardsdatascience.com/machine-learning-algorithms-part-9-k-means-example-in-python-f2ad05ed5203

[28] "Visualizing K-Means Clustering." [Online]. Available: https://www.naftaliharris.com/blog/visualizing-k-means-clustering/

[29] Y. Lu, I. Cohen, X. S. Zhou, and Q. Tian, "Feature selection using principal feature analysis," in *Proceedings of the 15th international conference on Multimedia - MULTIMEDIA '07*. Augsburg, Germany: ACM Press, 2007, p. 301. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1291233.1291297

[30] S. Solorio-Fernández, J. A. Carrasco-Ochoa, and J. F. Martínez-Trinidad, "A new hybrid filter–wrapper feature selection method for clustering based on ranking," *Neurocomputing*, vol. 214, pp. 866–880, Nov. 2016. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0925231216307718

[31] X. He and P. Niyogi, "Locality Preserving Projections," in *Advances in Neural Information Processing Systems 16*, S. Thrun, L. K. Saul, and B. Schölkopf, Eds. MIT Press, 2004, pp. 153–160. [Online]. Available: http://papers.nips.cc/paper/2359-locality-preserving-projections.pdf

[32] X. He, D. Cai, and P. Niyogi, "Laplacian Score for Feature Selection," in *Advances in Neural Information Processing Systems 18*, Y. Weiss, B. Schölkopf, and J. C. Platt, Eds. MIT Press, 2006, pp. 507–514. [Online]. Available: http://papers.nips.cc/paper/2909-laplacian-score-for-feature-selection.pdf

[33] D. Kolikov, "danilkolikov/fsfc," Mar. 2020, original-date: 2018-02-13T11:38:04Z. [Online]. Available: https://github.com/danilkolikov/fsfc

[34] "2.3. Clustering — scikit-learn 0.23.1 documentation." [Online]. Available: https://scikit-learn.org/stable/modules/clustering.html#clustering-performance-evaluation

[35] L. S. Sparke and J. S. Gallagher, *Galaxies in the universe: an introduction*, 2nd ed. Cambridge ; New York: Cambridge University Press, 2007, oCLC: ocm74967110.

[36] K. Glazebrook, C. Blake, F. Economou, S. Lilly, and M. Colless, "Measurement of the star formation rate from Ha in field galaxies at z = 1," p. 14, 1999.

[37] "Density-Based Clustering," library Catalog: blog.dominodatalab.com. [Online]. Available: https://blog.dominodatalab.com/topology-and-density-based-clustering/

[38] "TOPCAT." [Online]. Available: http://www.star.bris.ac.uk/~mbt/topcat/

[39] J. P. Huchra, J. P. Brodie, N. Caldwell, C. Christian, and R. Schommer, "Extragalactic Globular Clusters. IV. The Data," *The Astrophysical Journal Supplement Series*, vol. 102, p. 29, Jan. 1996. [Online]. Available: http://adsabs.harvard.edu/doi/10.1086/192250

A. Appendix

i. Database construction

The programming language of choice is Python. While programs like TOPCAT [38] are excelent for inspecting .fits tables and getting an understanding of the data we are dealing with, it lacks the adaptability and programming options that python offers. Using python also gives us access to a wide array of data science, machine learning and plotting modules.

To get started in python, we used the both astropy and pandas modules to load our .fits file and convert it into a pandas DataFrame object. Pandas DataFrames are two-dimensional data structures, which have labeled rows and colomns and offer a wide arange of analysis and plotting options. Wherever possible, we work with these DataFrames, on ocasion converting back to numpy arrays when so required. We can however export our databases as .fits and .csv files, but in general build up supporting databases in hd5 format.

```
import pandas as pd
from astropy.table import Table
def pandafy(fits_filename):
dat = Table.read(fits_filename, format='fits')
df = dat.to_pandas(index = 'CATAID')
return(df)
```

```
df = pd.merge(phot,spec,right_index=True, left_index=True, how='
    inner')
```

ii. Dataset features

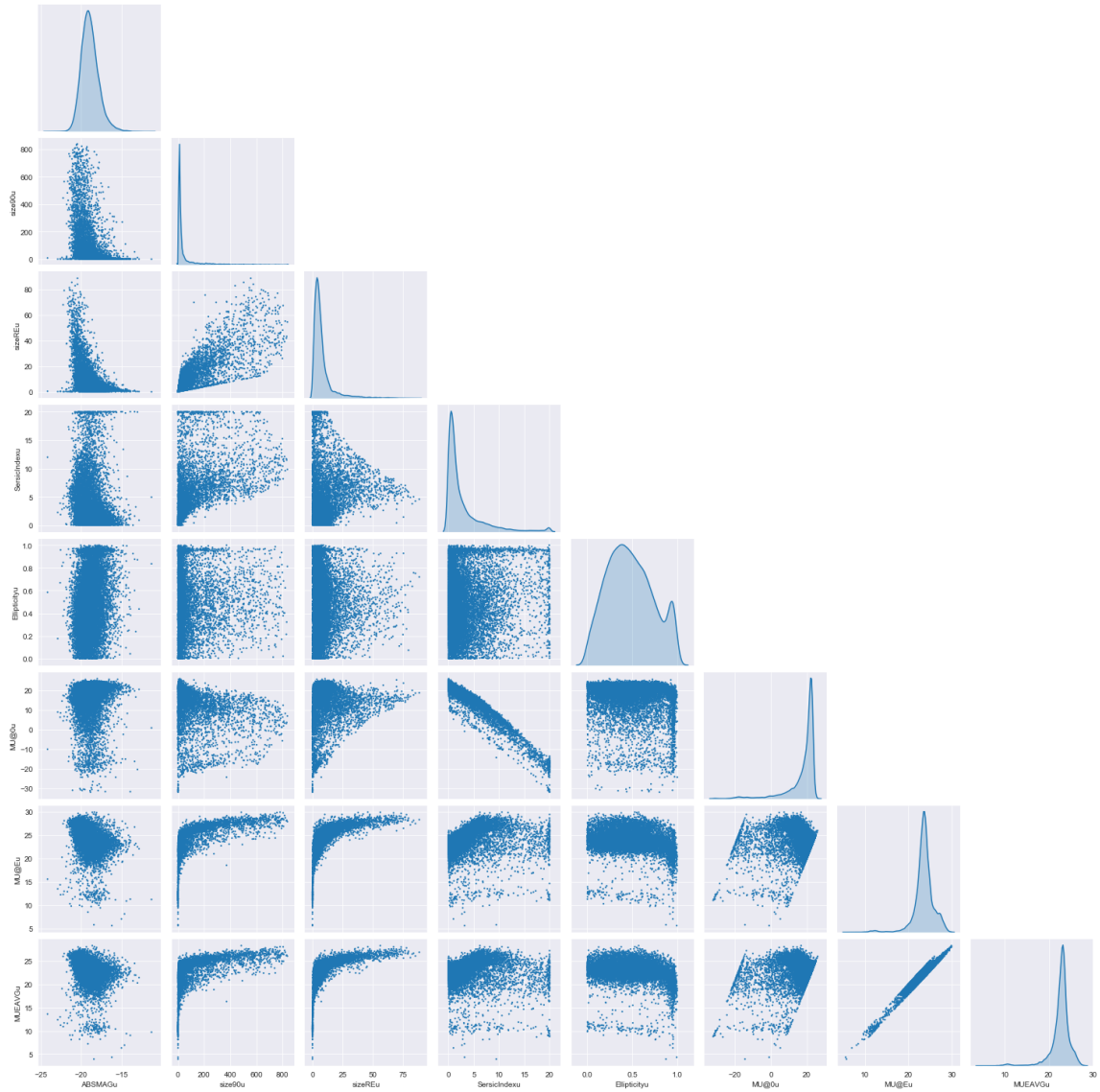


Figure 30: *Pairwise display of selected and filtered features in the u band. Every point is a galaxy. Most plots are scatterplots, while the diagonal hosts Kernel Density Estimations.*

Name	Type	Blo	Bhi	Llo	Lhi	Rlo	Rhi	description
OII	2	3667.000	3717.000	3717.000	3737.000	3737.000	3777.000	>OII 3727
HdA	2	4041.600	4079.750	4083.500	4122.250	4128.500	4161.000	>Hdelta A (Worthey & Ottaviani 1997)
HdF	2	4057.250	4088.500	4091.000	4112.250	4114.750	4137.250	>Hdelta F (Worthey & Ottaviani 1997)
CN1	1	4080.125	4117.625	4142.125	4177.125	4244.125	4284.125	>Lick
CN2	1	4083.875	4096.375	4142.125	4177.125	4244.125	4284.125	>Lick
Ca4227	2	4211.000	4219.750	4222.250	4234.750	4241.000	4251.000	>Lick
G4300	2	4266.375	4282.625	4281.375	4316.375	4318.875	4335.125	>Lick
HgA	2	4283.500	4319.750	4319.750	4363.500	4367.250	4419.750	>Hgamma A (Worthey & Ottaviani 1997)
HgF	2	4283.500	4319.750	4331.250	4352.250	4354.750	4384.750	>Hgamma F (Worthey & Ottaviani 1997)
HgVA125	1	4330.000	4340.468	4333.000	4352.737	4359.250	4368.750	>Hgamma (Vazdekis & Arimoto 1999), sigma=125
HgVA200	1	4331.000	4340.750	4332.000	4352.250	4359.250	4368.750	>Hgamma (Vazdekis & Arimoto 1999), sigma=200
HgVA275	1	4331.500	4341.000	4331.500	4351.875	4359.250	4368.750	>Hgamma (Vazdekis & Arimoto 1999), sigma=275
OIII4363	2	4283.500	4319.750	4355.000	4371.000	4371.000	4419.750	>4364 Owers def.
Fe4383	2	4359.125	4370.375	4369.125	4420.375	4442.875	4455.375	>Lick
Ca4455	2	4445.875	4454.625	4452.125	4474.625	4477.125	4492.125	>Lick
Fe4531	2	4504.250	4514.250	4514.250	4559.250	4560.500	4579.250	>Lick
Fe4668	2	4611.500	4630.250	4634.000	4720.250	4742.750	4756.500	>Lick
Hb	2	4827.875	4847.875	4847.875	4876.625	4876.625	4891.625	>Lick
Hb_p	2	4815.000	4845.000	4851.320	4871.320	4880.000	4930.000	>Hbeta plus from Gonzalez thesis (p116)
OIIIB	2	4885.000	4935.000	4948.920	4978.920	5030.000	5070.000	>OIII_1 from Gonzalez thesis (p116)
OIIIR	2	4885.000	4935.000	4996.850	5016.850	5030.000	5070.000	>OIII_2 from Gonzalez thesis (p116)
Fe5015	2	4946.500	4977.750	4977.750	5054.000	5054.000	5065.250	>Lick
Mg1	1	4895.125	4957.625	5069.125	5134.125	5301.125	5366.125	>Lick
Mg2	1	4895.125	4957.625	5154.125	5196.625	5301.125	5366.125	>Lick
Mgb5177	2	5142.625	5161.375	5160.125	5192.625	5191.375	5206.375	>Lick
Fe5270	2	5233.150	5248.150	5245.650	5285.650	5285.650	5318.150	>Lick
Fe5335	2	5304.625	5315.875	5312.125	5352.125	5353.375	5363.375	>Lick
Fe5406	2	5376.250	5387.500	5387.500	5415.000	5415.000	5425.000	>Lick
Fe5709	2	5672.875	5696.625	5696.625	5720.375	5722.875	5736.625	>Lick
Fe5782	2	5765.375	5775.375	5776.625	5796.625	5797.875	5811.625	>Lick
NII5755	2	5710.000	5740.000	5747.000	5763.000	5765.000	5795.000	>Owers def.
Na5895	2	5860.625	5875.625	5876.875	5909.375	5922.125	5948.125	>Lick
TiO1	1	5816.625	5849.125	5936.625	5994.125	6038.625	6103.625	>Lick
TiO2	1	6066.625	6141.625	6189.625	6272.125	6372.625	6415.125	>Lick
OI6300	2	6250.000	6290.000	6292.000	6308.000	6320.000	6350.000	>Owers def.
SIII6312	2	6250.000	6290.000	6306.000	6320.000	6320.000	6350.000	>Owers def.
OI6364	2	6320.000	6350.000	6356.000	6372.000	6375.000	6405.00	>Owers def.
Ha	2	6490.000	6530.000	6553.000	6573.000	6600.000	6650.000	>6562.80
NIIB	2	6490.000	6530.000	6541.000	6555.000	6600.000	6650.000	>6547.96
NIIR	2	6490.000	6530.000	6576.000	6591.000	6600.000	6650.000	>6583.34
SIIB	2	6650.000	6705.000	6710.000	6723.500	6745.000	6800.000	>6716.31; Owers def.
SIIR	2	6650.000	6705.000	6723.500	6737.500	6745.000	6800.000	>6730.68; Owers def.
BH_CNB	1	3785.000	3810.000	3810.000	3910.000	3910.000	3925.000	see caption
BH_HK	1	3910.000	3925.000	3925.000	3995.000	3995.000	4010.000	>as for BH_CNB
BH_CaI	1	4200.000	4215.000	4215.000	4245.000	4245.000	4260.000	>as for BH_CNB
BH_G	1	4275.000	4285.000	4285.000	4315.000	4315.000	4325.000	>as for BH_CNB
BH_Hb	1	4800.000	4830.000	4830.000	4890.000	4890.000	4920.000	>as for BH_CNB
BH_MgG	1	5125.000	5150.000	5150.000	5195.000	5195.000	5220.000	>as for BH_CNB
BH_MH	1	4740.000	4940.000	4940.000	5350.000	5350.000	5550.000	>as for BH_CNB
BH_FC	1	5225.000	5250.000	5250.000	5280.000	5280.000	5305.000	>as for BH_CNB
BH_NaD	1	5835.000	5865.000	5865.000	5920.000	5920.000	5950.000	>as for BH_CNB

Table 5: *Direct Summation Subset contents.[18]*
Direct summation line bands, where Blo, Bhi, Llo, Lhi and Rlo and Rhi are the limits of the blue continuum band, line band and red continuum band.
Type 1 = molecular line, Type 2 = EW
For BH_CNB, see [39].

iii. Isolation Forest

The average pathlength for all data points in a tree is:

$$c(n) = 2H(n-1) - (2(n-1)/n) \tag{5}$$

Where $c(n)$ is the average pathlength across the entire tree, n is the number of data entries in our sample and $H(i)$ is the harmonic number, which can be estimated by $\ln(i) + 0.5772156649$ (Euler's constant). This can be used to define the anomaly score by:

$$s(x,n) = 2^{-\frac{E(h(x)))}{c(n)}} \tag{6}$$

where $s(x,n)$ is the anomaly score, $E(h(x))$ is the average of the pathlength $h(x)$ for a specific datapoint x and $c(n)$ is defined above.

If the values are much smaller then 0.5, then it's quite safe to regard objects as normal. Conversely, as values increase further and further above 0.5, they are increasingly likely to be anomalies, with objects that have an s very close to 1 being anomalies. Depending on the strictness, any objects with a value above 0.6 are likely anomalies.

Contrary to most other methods, isolation trees work best when the sample size is kept small. Because it takes a new sample for every tree, it keeps the effects the swamping and masking of anomalies This is due to effects called swamping and masking. Swamping is when anomalies are close to normal objects, making them harder to detect. Masking is when the existence of too many anomalies within the sample starts to conceal their own presence. Isolation Forest use of sub-sampling fixes these problems, however, controlling the data size of each individual tree by spreading the total sample out over many trees and allowing each individual tree to be specialised due to including a different set of anomalies or even no anomalies at all.
Another important characteristic and the reason I settled on an Isolation Tree method is the performance. This algorithm performs excellent especially when applied to large datasets with a lot of data points. The complexity in big $\mathcal{O}$ notation (A standard for indicating the speed and memory requirement of an algorithm)for the training step is $\mathcal{O}(t\psi \log \psi)$ and for the evaluation stage: $\mathcal{O}(n\psi \log \psi)$, where ψ is the sample size, t the number of trees, and n the total number of data points, far lower than some Anomaly Detection methods which rely on calculating distances between points for example (Which can go to $\mathcal{O}(2n^2)$ or can require system memory of $(2n)^2$.

Other then this, Isolation forests are not only fast, but also accurate. This accuracy is measured in terms of AUC, which stands for Area Under the ROC Curve. A ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model. This curve plots two parameters against each other, the True Positive Rate and the False Positive Rate. The Area Under this Curve can range from 0 to 1, where 1 means that the model functions perfectly, 0.5 means it can't distinguish class at all, and 0 implies that it inverts the classes and in our case consistently considers outliers to be normal and normal points to be anomalies.

Correlation matrixes

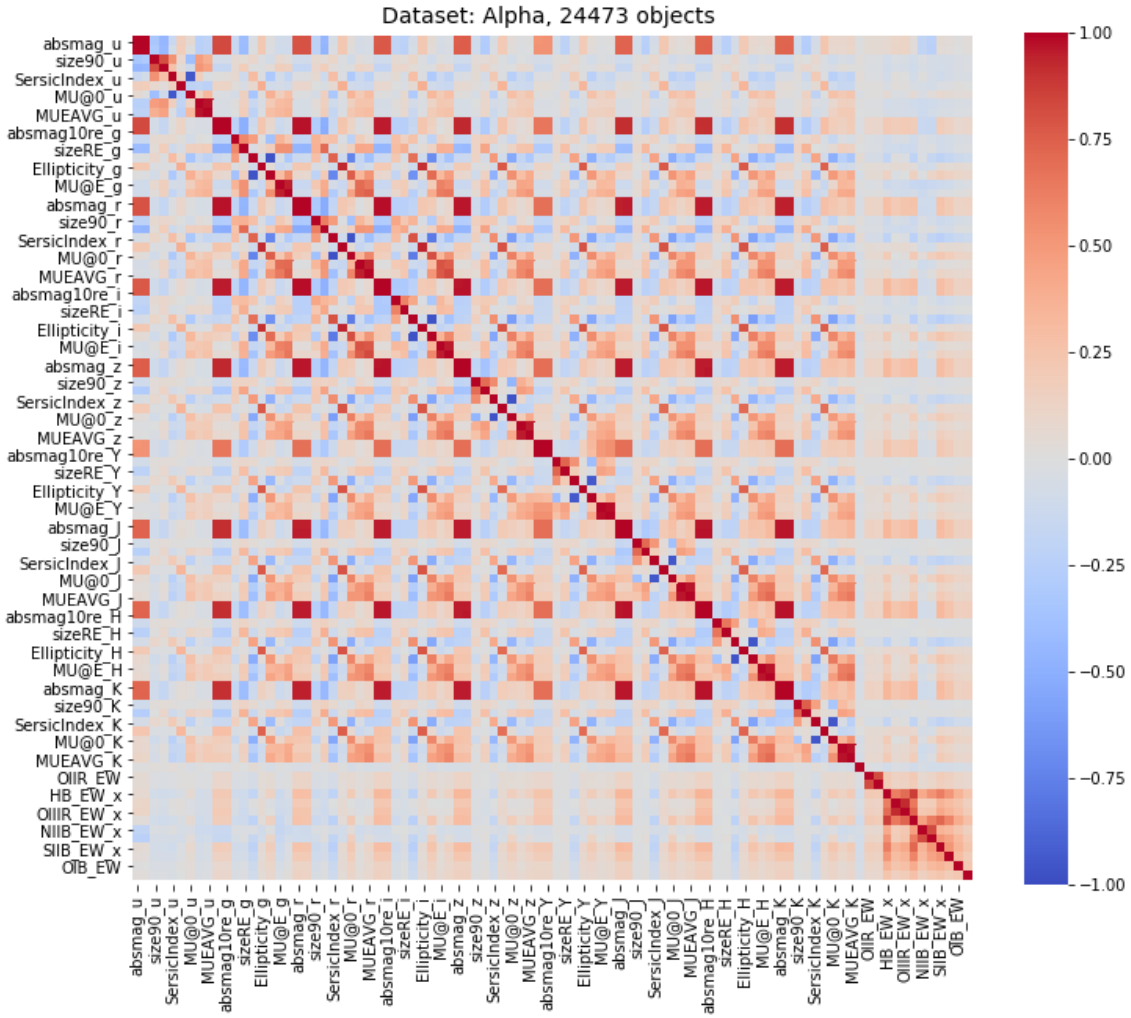


Figure 31: *Correlation matrix for whole dataframe*

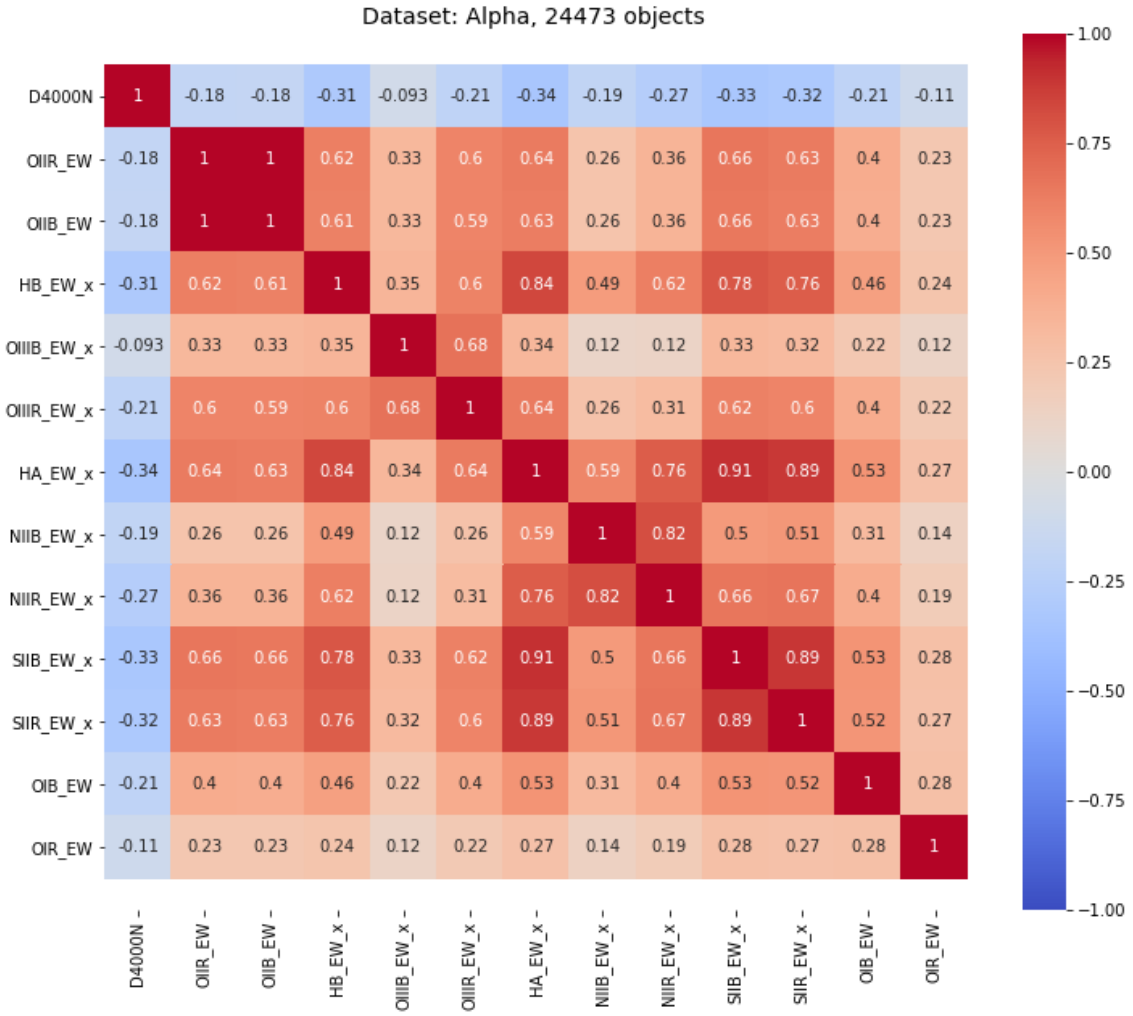
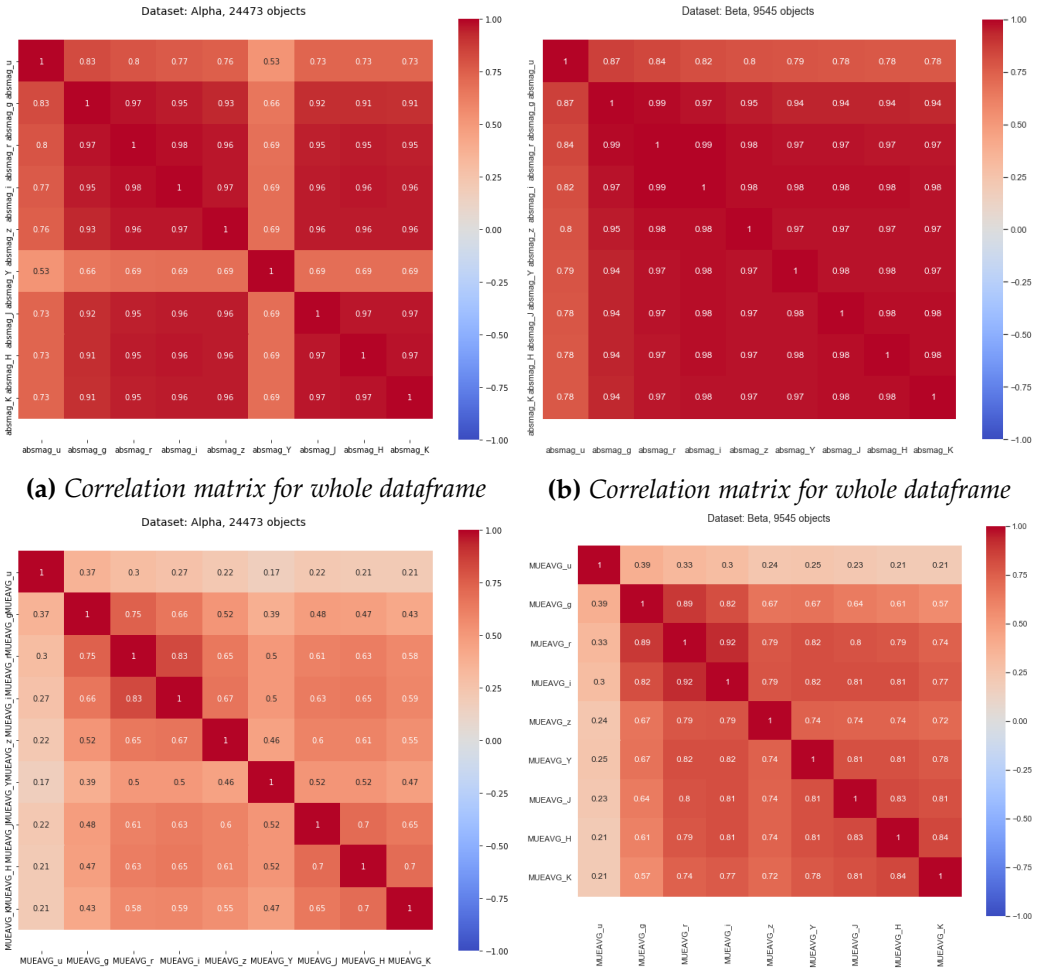


Figure 32: *Correlation Matrix for Spectral part*



(a) *Correlation matrix for whole dataframe*

(b) *Correlation matrix for whole dataframe*

(c) *Correlation matrix for whole dataframe*

(d) *Correlation matrix for whole dataframe*

Figure 33: *Correlation matrix examples for the Alpha dataset. Any numbers given correspond to the strength of the correlation*