University of Sussex
Department of Physics & Astronomy

Using Machine Learning Algorithms to Identify Fast and Slow Rotating Galaxies

Joshua Fenech

Submitted in part fulfilment of the requirements for the degree of
Physics with Astrophysics

Abstract

Decision trees were successfully applied to the photometric variables seŕsic index, n , which models the galactic profile as $I(R) = I_e exp\{-b_n[(R/R_e)^{1/n} - 1]\}$, and D/T ratios, the proportion of light from the disk component to the total light of the galaxy, for the subset of 260 ATLAS$_{3D}$ galaxies for which the spin parameter λ_{Re} had been calculated by Emsellem et al. [4] and photometric quantities determined by Krajnovic et al[6]. When trained using these variables independently using the initialised parameter options, the results were promising, with success rate based on n reaching 73%. When trained using D/T alone, the success was increased to 81%, but this result was skewed by the large number of galaxies with no exponential disk component, where D/T$\lesssim$0.05, being classed universally as fast rotators. When both parameters were used for training, the algorithm did marginally better than with n alone, achieving a success rate of 75%, where more variation was seen in estimating the spin of galaxies with no disk component, but did so incorrectly. The most promising results were achieved when the inbuilt sklearn function GridSearchCV was used that exhaustively searches the parameter space over a given range of values and/or options, and partitions the data into k subsets and recursively trains on k-1 subsets retaining 1 for validation of results. In so doing, the best parameters were found and the success rate increased dramatically to 93%.

Acknowledgements

Data was extracted from from [3] and [6] principally. I performed all of the data analysis and plots except for those which are referenced. The code utilised metrics and algorithms from the Scikit-Learn module. Pandas were also used for data handling and for the scatter matrices. Matplotlib was used for the majority of plots. I would also like to thank my supervisor for the extended discussions we had over the duration of the project and his endless knowledge.

Contents

List of Tables

List of Figures

Chapter 1

Background Theory

1.1 Introduction

Galaxy morphology has traditionally been classified based on the Hubble sequence, originating from the identification galaxy features from photographic plates. However, this classification is based on visual distinctions and fails to accurately represent early-type galaxies (E's and S0's) and it was argued by [3] and [4] that a more telling classification would be based on the spin parameter due to the intrinsic qualitative change in velocity structure exhibited by galaxies, with a threshold of separating slow ($\lambda < 0.1$) and fast (≥ 0.1) rotators, as can be seen below. This threshold was later updated to include the ellipticity ϵ by defining slow rotators and fast rotators have λ_{Re} lower and larger than $k_{FS} \times \sqrt{\epsilon}$, respectively, where $k_{FS} = 0.31$ for measurements made within an effective radius R_e [4]. This new criterion is nearly independent of viewing angle. λ_R is defined as[1]:

$$\lambda_R = \frac{\sum_{i=1}^{N_p} F_i R_i |V_i|}{\sum_{i=1}^{N_p} F_i R_i \sqrt{V_i^2 + \sigma_i^2}} \tag{1.1}$$

where F_i, R_i, V_i and $_i$ are the flux, circular radius, velocity and velocity dispersion of the ith spatial bin, the sum running on the N_p bins. λ_{Re} indicates spin parameter calculated within 1 effective radius R_e. This is superior over a velocity dispersion classification, V/σ, which fails when confronted by galaxies with kinematically decouple cores (KDC), "whose angular momentum vector is misaligned with respect to that of the bulk of the galaxy" [2]. Furthermore, λ_R 'can quantify galaxy morphology via the kinematic properties of galaxies[7]', beyond early types.

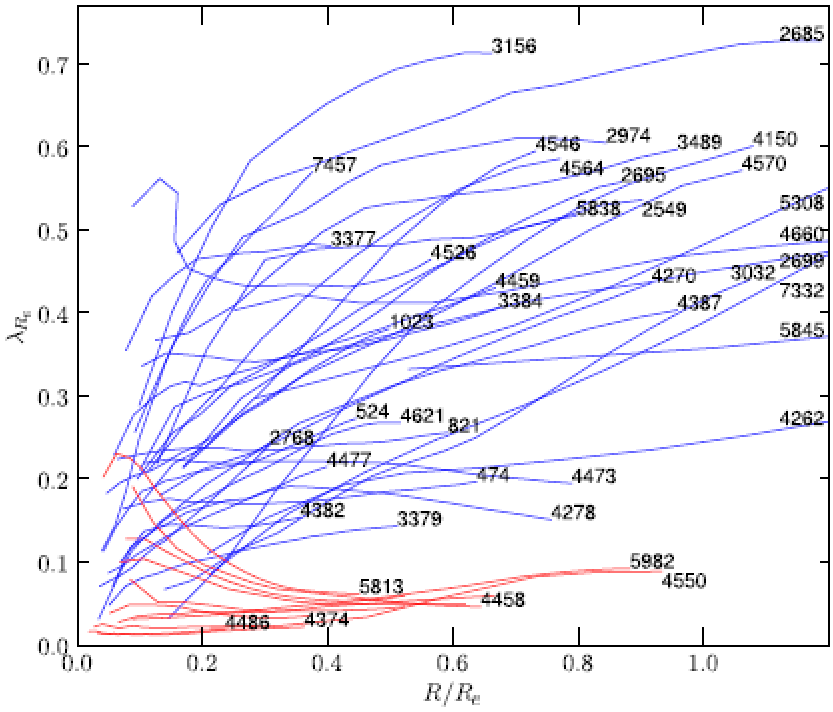


Figure 1.1: Radial λ_R profiles for the 48 E and S0 galaxies of the SAURON sample. Profiles of slow and fast rotators are coloured in red and blue, respectively. NGC numbers are indicated for all fast rotators and most slow. rotators [4]

As can be seen in figure 1.2, there appears a weak correlation for galaxies in the ATLAS$_{3D}$ survey for early-type galaxies, suggesting that based on seŕsic index alone there is some prospect for applying statistical methods to infer morphology. The spin parameter is costly to determine due to its reliance on integral field spectroscopy and has therefore only been found for a small number of galaxies: 260 from the ATLAS3D survey and 446 from SAMI. This compares with over 500 million galaxies with photometric data from the Sloan Digital Sky Survey (SDSS) alone [12]. Although other classifications do not rely on this parameter, it is still of value in relation to other properties. It is therefore of great value to find an alternative means of identifying the rotation. Although the motivation for this study is to extend identifying the spin parameter to all morphological classes, this paper will focus on the early-type galaxies taken from [4], defined

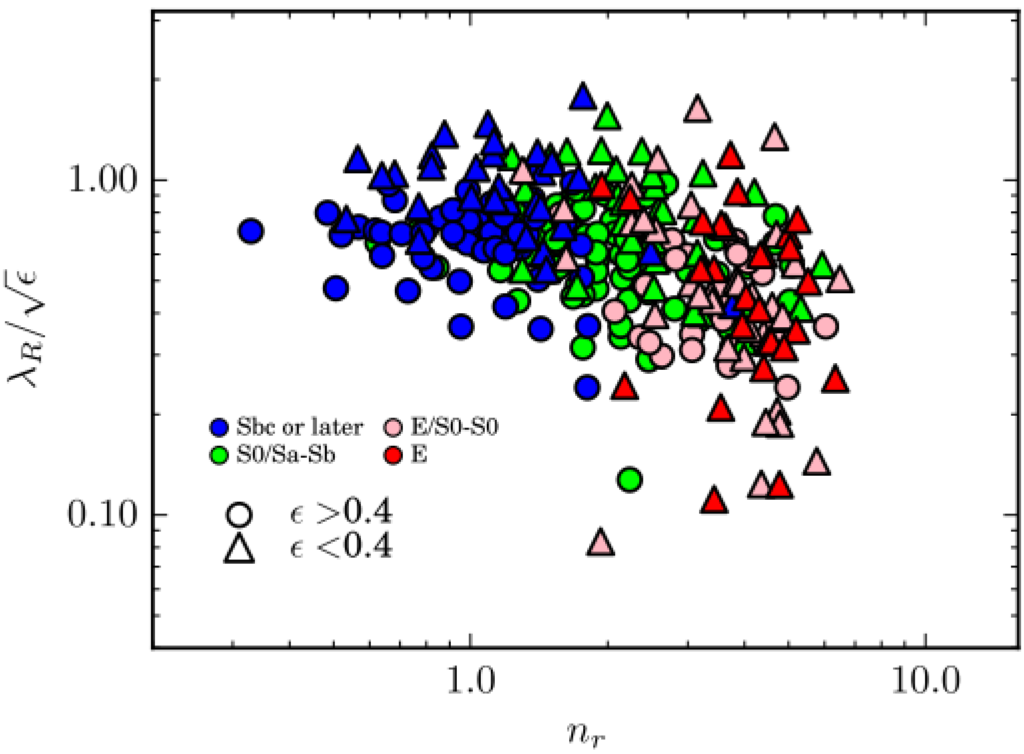


Figure 1.2: Galaxies from ATLAS$_{3D}$ colour coded by optical morphology. There appears a weak correlation for early types that grows more pronounced for late types (Sbc or later), using seŕsic index as a weak proxy for morphology. [7]

as Hubble type E/S0's, T-type $T < 3.5$ (Es) and $T \geq 3.5$ (S0s).

1.2 Variations of the Spin Parameter

The ATLAS3D paper measured λ to 1 effective radius, R_e and to half the effective radius, $R_e/2$, where

$$I(R_e = I_0/e) \tag{1.2}$$

The spin parameter was later updated by [3] to take account of possible line-of-sight dependencies on the orientation by including the ellipticity, and thus was defined as:

$$\lambda_{Re} = \frac{\lambda}{\sqrt{\epsilon}} \tag{1.3}$$

and hence the threshold for separating fast and slow rotators was defined as:

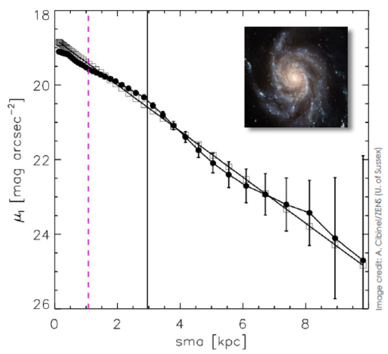
$$\lambda_{Re} = (0.31 \pm 0.01) \times \sqrt{\epsilon} \tag{1.4}$$

1.3 Sérsic Index of the Single Fit and Bulge Component, n and n_b

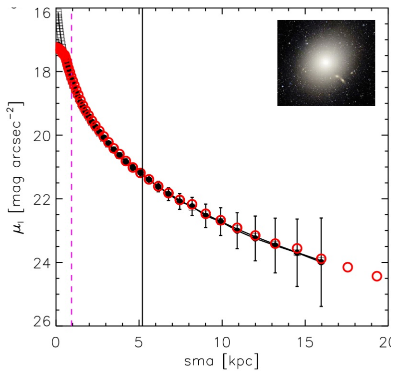
The Sérsic profile models the light intensity over the surface of the galaxy in terms of an exponential function as a function of the distance from the centre, R, and the Sérsic index n:

$$I(R) = I_e exp\{-b_n[(R/R_e)^{1/n} - 1]\} \tag{1.5}$$

As such, this parameter is one way of quantifying the morphological differences between structurally distinct galaxies. The light will vary dependent on the different populations of stars and their distribution throughout the galaxy.



(a) Late-type Spiral Galaxy



(b) Early-Type Elliptical Galaxy

Figure 1.3: Seŕsic Index as a Descriptor of Morphology

1.4 Disk-to-Total Light Ratio, D/T

This measures the proportion of light from the galaxy that emanates from the disk compared to the total light measured from the galaxy.

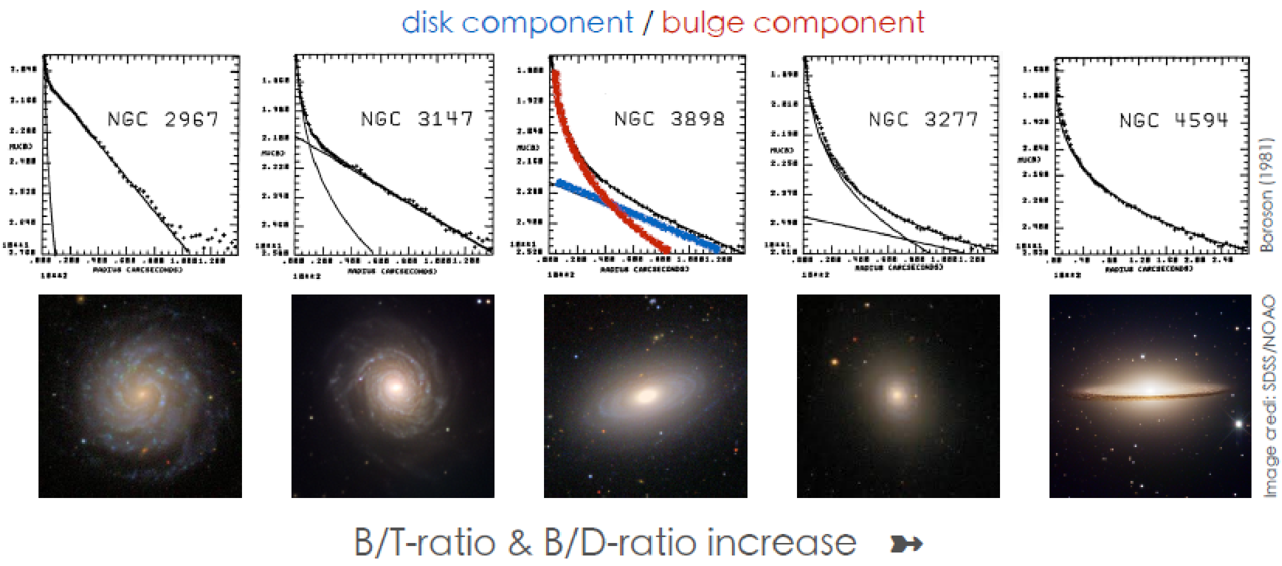


Figure 1.4: The D/T ratio measures how much of a galaxies light is distributed throughout the disk compared to the total light. Late-types have more pronounced spiral arms and so have a larger D/T value. [9]

1.5 ATLAS^{3D}

According to [3], this survey focused on a 'volume-limited ($1.16 \times 10^5 Mpc^3$) sample of 260 early-type (elliptical E and lenticular S0) galaxies (ETGs)...The sample consists of nearby (D $<$ 42 Mpc, $|\delta - 29°| < 35°$, $|b| > 15°$) morphologically selected ETG's extracted from a parent sample of 871 galaxies (8 per cent E, 22 per cent S0 and 70 per cent spirals) brighter than $M_K < -21.5$mag (stellar mass $M_\star \gtrsim 6 \times 10^9 M_\odot$).' ETG's were defined as having de Vaucouleurs T type T $>$ -3.5 and T ≤ -3.5 for S0 and E galaxies respectively, which correlates with the Hubble classes lenticular and elliptical.

Chapter 2

Methods

2.1 The scikit-learn library

It was chosen to implement algorithms from the scikit-learn module of the python language since it "exposes a wide variety of machine learning algorithms, both supervised and unsupervised, using a consistent, task-oriented interface, thus enabling easy comparison of methods for a given application" [5]. This allowed several different algorithms to be implemented within the same environment and gain meaningful results with minimal prior coding and analysis. The algorithms initially chosen was decision trees (DT's) since these allow both regression and classification analysis, and are known as 'white boxes' due to their relatively transparent process whereby the mechanics of training could be evaluated more readily.

2.2 Decision Trees

Different machine learning algorithms use different statistical procedures in order to evaluate data and make inferences. DT's emulate a logical classification procedure similar to that used in biology to identify species. Starting from the full dataset a series of binary if-then tests are consequentially performed and data split into 2 branches continuously until a final statistical criterion is satisfied. The algorithm does this by arbitrarily choosing a value that splits the parameter space in 2, forming a node and 2 branches. A statistical test (purity measure), either Gini impurity or entropy of information gain, is then applied to the 2 child datasets separately to evaluate how correlated the members within each child are, and the resulting values summed proportionately to find the total purity. Decision trees then recurse through the parameter space in a greedy fashion to find those that maximise the purity, and selects this for the split. Once found, the process repeats on each of the child datasets until a predefined stopping criterion is fulfilled, producing a leaf at the end of each branch. This results in a series of rules that can be applied to unseen data in an attempt to reproduce the classifications. The Gini impurity is defined as:

$$H(X_m) = \sum_k p_{mk}(1 - p_{mk}), \tag{2.1}$$

and the entropy defined as:

$$H(X_m) = -\sum_k p_{mk} \log p_{mk}, (2.2)$$

where the target classification takes on value $0, 1, .., k-1$ for node m, representing a region R_m with N_m observations. For a single input variable x_i with class label y_i (i.e. fast/slow rotation), the proportion of class k observations in node m is given by:

$$p_{mk} = 1/N_m \sum_{x_i \subset R_m} I(y_i = k) \tag{2.3}$$

In the case of entropy, the purity is measured in terms of entropy gain, defined as the entropy of the parent minus the entropy of the child.

The benefits of decision trees are that they are more interpretable, employing a 'white box' approach, compared to other algorithms (i.e. neural nets) that have a relatively opaque process. They require little data preparation compared to others that need, for example, normalisation, and are well suited to binary decision classes, as well as being fast. They are also able to handle trends of different order (linear, polynomial etc.). Disadvantages are that they can result in overfitting (an example of which can be seen in 2.1), where the rules found are overly complex

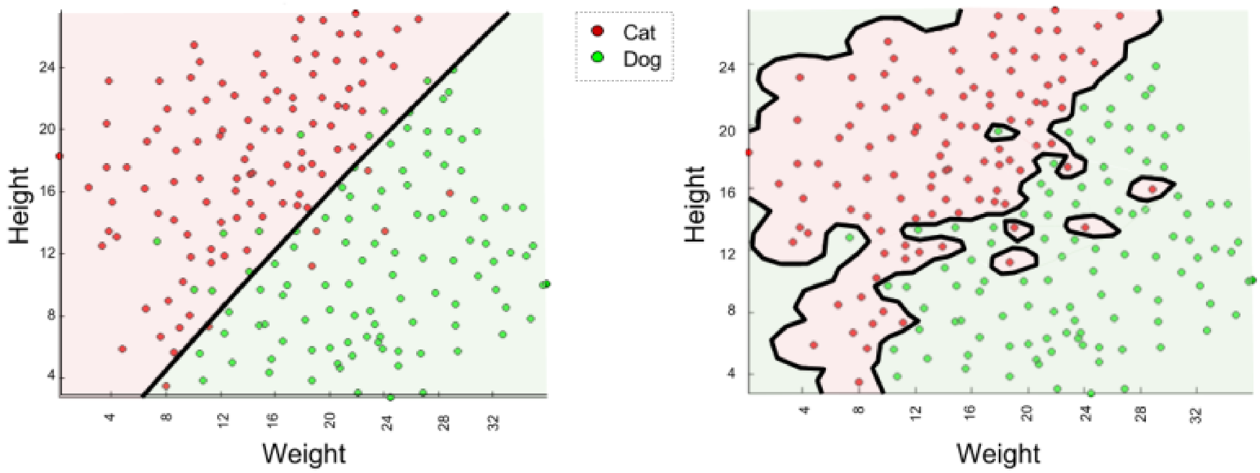


Figure 2.1: Demonstration of how the same data can be fit in different ways. The example on the left sacrifices some purity for simplicity, but represents the global trend more accurately[11].

and specific to the training set and do not generalise to unseen data. There are, however, methods to minimise this by specifying the minimum number of samples in a subset required to form a branch, or the maximum number of nodes allowed (pruning methods, which look to minimise this affect after the tree has been trained, are not supported in SKLearn). Another problem lies in the fact that locally optimal decisions are made at each point, which may fail to recognise more global patterns. Furthermore, because of the hierarchical nature of data fitting, a small change in the data can result in drastically different results as the rule changes promulgate through the iterative decision process. The backend of the SKlearn modules were not evaluated directly and implemented naïvely. The parameters were initialised to their defaults as described in the documentation[10]. Using the modules themselves required elementary use of python to pass the module arrays of necessary values. Implementation proceeded in 2 stages once the data had been suitably formatted. The data was arbitrarily split into 2 groups, a training and a test set, based solely on their position within the results: they were ordered according to their LEDA classification and occupied different locations of the sky, and so should not have any bias. For classification, the training set was passed to the module as a list with each item holding the feature value, (i.e. Sérsic index) or a list of feature values when more than one feature used, and target variable (fast/slow rotator classification). A function was output that incorporated the learned rules which was then applied to the test data set resulting in an array of predicted values that could be measured against the known values. The algorithm was run as a classification problem due to the more simple analysis of success and errors, without recourse to analysing more complex error distributions.

2.2.1 Parameters

SKLearn allows several parameters (outlined in Table 2.1) to be adjusted manually in order to maximise the efficiency of the models, and these are particular to each algorithm.

2.3 Data & Formatting

2.3.1 ATLAS^{3D}

Spectroscopic data (namely Sérsic index of the single fit and bulge component, n and n_b respectively, and D/T, the Disk-to-Total light ratio) was extracted from [6] whilst the kinematic parameters λ_{Re}, $\lambda_{Re/2}$ and the Fast/Slow rotation classification were extracted from [4]. The data from both sources was combined using a pandas dataframe.

The classifier was first trained using the spin parameter λ_{Re} with the FS rotation classification as target variable as a test measure, which successfully predicted 100% of the test set. The data was then evaluated for statistical correlations between spectroscopic parameters and the spin parameter. This was performed using the inbuilt scatter matrix command of the pandas module that plots each variable against the other, with the diagonal used to depict the kernel density estimation (kde) which estimates the probability density function of the variable. This was performed for the two rotator populations individually.

As can be seen from the figures 2.3 and 2.2, there are no immediately obvious distinguishing features that separate the two populations. The R_{max} population peaks at around 0.8 with a much sharper peak for FR's compared to a broader distribution centred at 0.6 for SR's. The ellipticity for FR's exhibits a double peak at 0.2 and 0.6, values ranging from 0.0-0.9 while SR's exhibit a single peak centred at 0.2 but a smaller range of 0.0-0.5. The D/T ratios share very similar distributions but FR's have flatter maximums and minimums. The Seŕsic index

Table 2.1: SKLearn Decision Tree Parameters

Parameter	Options	Description
Criterion	Gini or Entropy	Uses the gini impurity as outlined above or entropy for information gain. The gini impurity default was used
Splitter	Best or Random	Chooses the best split or the best random split to make at each node based on the criterion.
max_features	int	Considers the maximum number of features of the dataset to consider when recursing for the best split.
max_depth	int or None	The maximum depth of the tree, i.e. how many splits the tree makes. The default of None will recurse until all leaves are pure, i.e. contain 1 class, or contain less than the min_samples_split.
min_samples_split	int,float (default=2)	The minimum number of samples required to be at a leaf node, where the leaf node represents the split into classifications.
max_leaf_nodes	int, none (default=None)	The maximum number of leaf nodes allowed, with the tree choosing nodes which best minimises the gini impurity.
min_weight_fraction_leaf	float or None	The minimum weighted fraction of the sum total of weights (of all the input samples) required to be at a leaf node. Samples have equal weight when sample_weight is not provided. No weights were supplied since and so were equally weighted.
min_impurity_split	float (default=1e-7)	Threshold for early stopping in tree growth. A node will split if its impurity is above the threshold, otherwise it splitting will cease and the node forms a leaf.
presort	bool, (default=False)	Option to possibly speed up training process, not used here.

distributions are almost identical for both populations with a peak at n=2 and an extended tail towards higher values, except that SR's have a significant number of galaxies with n=1, indicating a tendency to have a purely exponential profile, whilst the Seŕsic index of the bulge had a larger range of 0-11 compared to 0-9, but with a similar lineshape with peak at $n_b \approx$0.2 compared to peak at $n_b \approx$0.6. The flattening of the bulge component, q_b, had minimum and maximum at 0.25 and 0.75 respectively, but for the fast rotators there was a second smaller peak at 0.0. Both populations exhibited 2 peaks at 0.0 and 18.0 for the effective surface brightness, although the second peak at the higher value was more pronounced for SR's. The flattening of the exponential component q_d distributions were very similar in both cases, centred at 0.0, except for a shallower tail skewed to higher values in FR's. Both populations also had a peak centred at $\approx$21.

These plots indicate that there are few if any distinct distinguishing features by which the rotation classification can be categorically determined. The fact, however, that there are multiple variables that do vary slightly between the populations suggests that a machine learning algorithm could identify some empirical rules combining these traits to achieve such an aim.

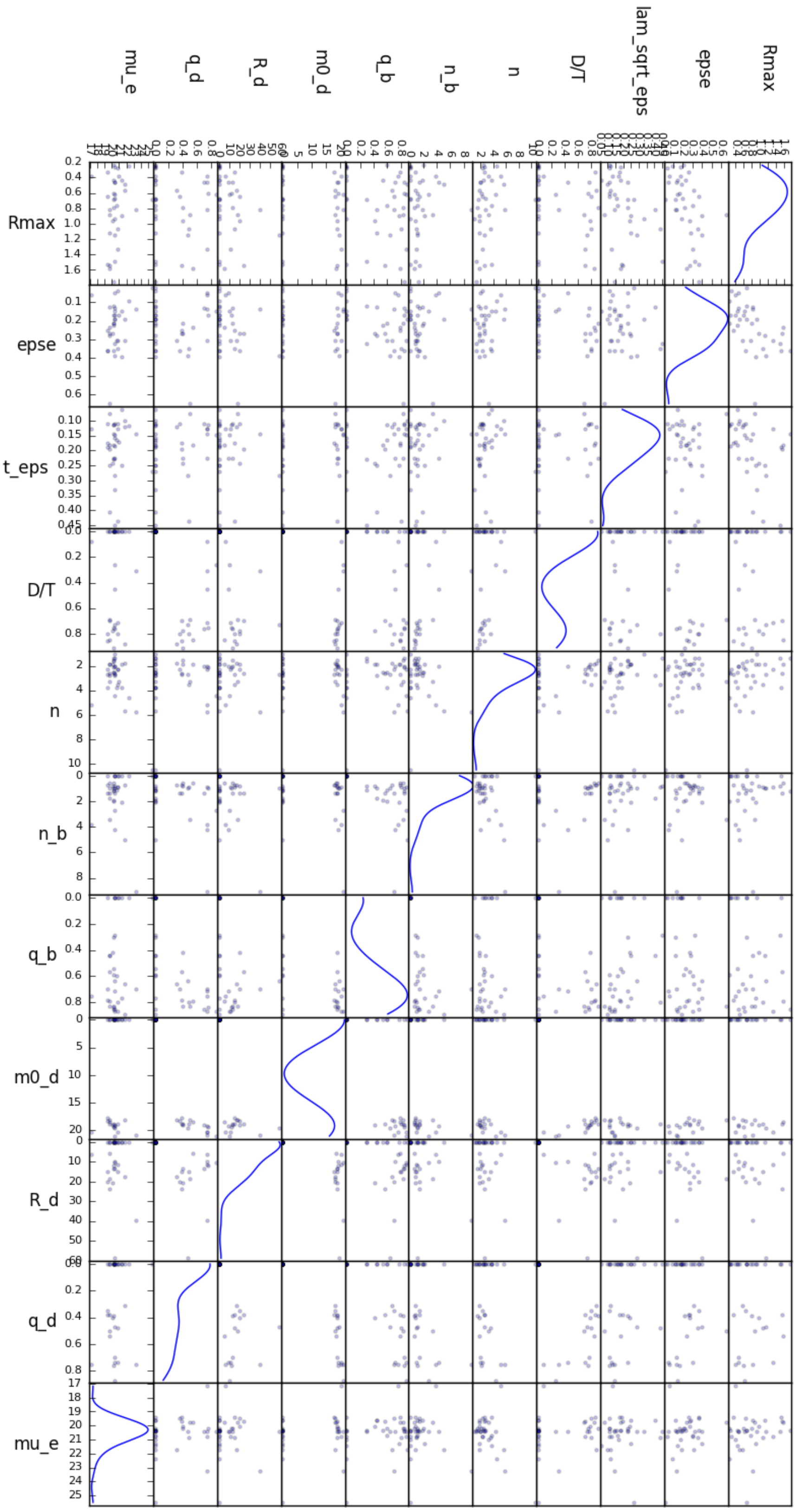


Figure 2.2: Scatter matrix of the slow rotator population

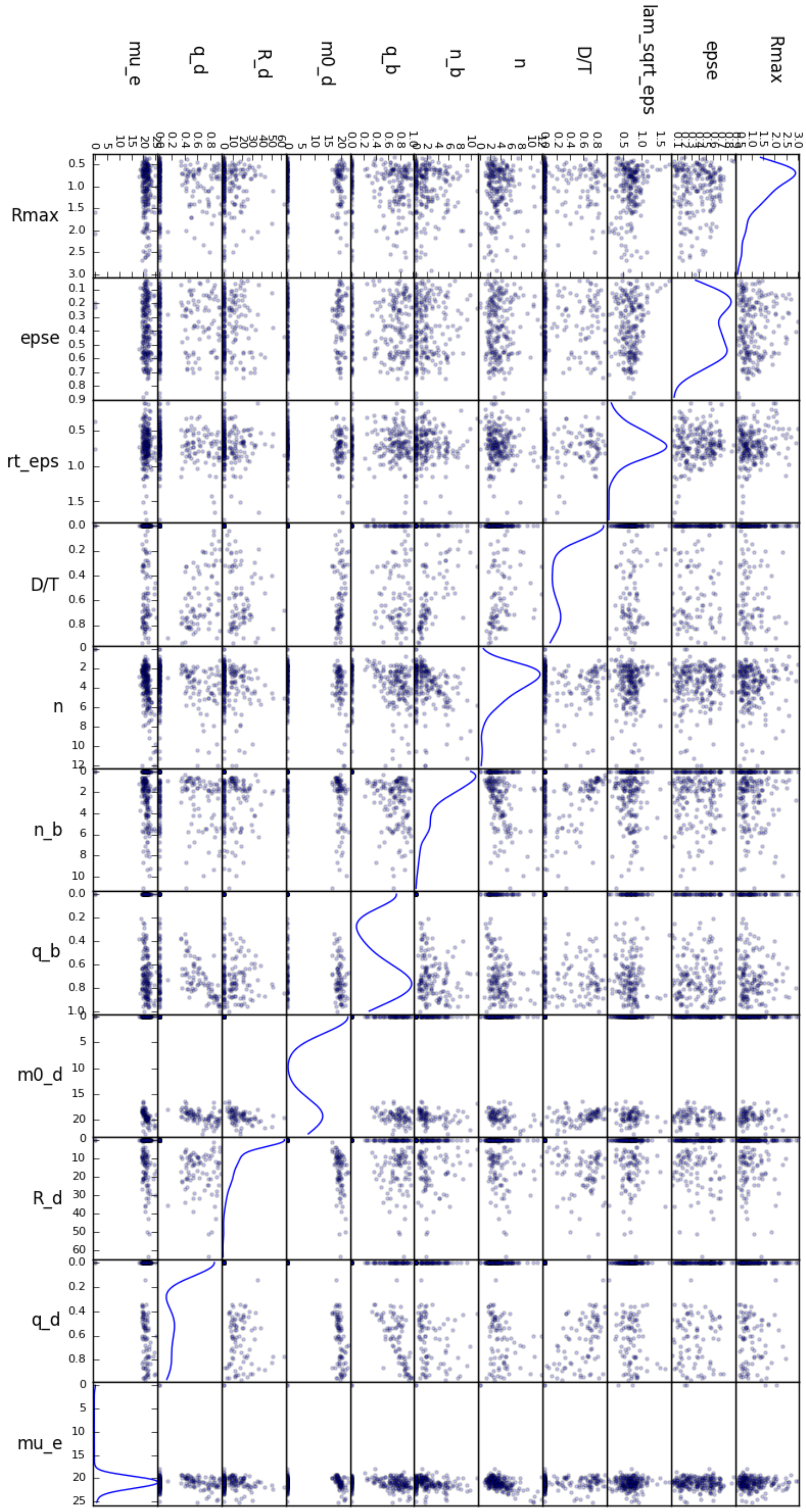


Figure 2.3: Scatter matrix of the fast rotator population

Chapter 3

Results & Conclusions

3.1 ATLAS$^{\mathbf{3D}}$

Initially, the algorithm was trained using the λ_{Re} to predict the FS classification since they were related directly. Promisingly, the classifier was 100% successful in its predictions. Training was then performed using the Seŕsic index of the single fit, n, and achieved success of 71%. We can estimate how the model succeeded for the separate populations compared to that predicted by the binomial probability distribution which is necessary due to the binomial nature of the outcome with differing sizes of the two populations, given by[8]:

$$P(k, n) = \binom{n}{k} p^k q^{n-k} \tag{3.1}$$

where n is the number of trials, k is the number of successes, $n - k$ is the number of failures, p is the probability of success in one trial and $q = 1 - p$ is probability of failure in one trial. Since the binomial distribution does not take account of the order of the successes, we adapt the binomial formula by removing the permutations factor:

$$P(k, n) = p^k q^{n-k} \tag{3.2}$$

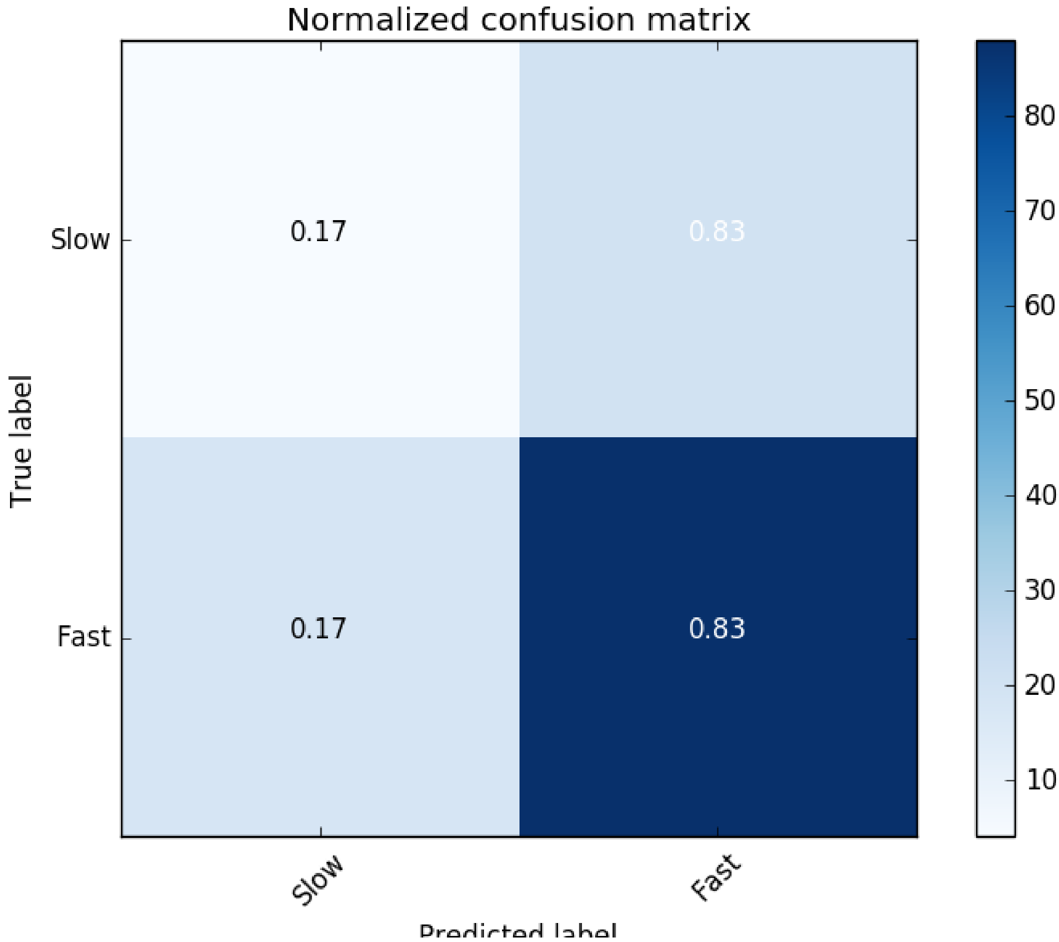


Figure 3.1: Confusion Matric for predictions based on Seŕsic Index.

By treating the 2 populations separately we can therefore define success as a correct classification, and the probabilities are given by the overall relative number proportions of the 2 populations, i.e.,

$$P_f(k_f, n_f) = p_f^{k_f} q_f^{n_f - k_f} \tag{3.3}$$

and,

$$P_s(k_s, n_s) = p_s^{k_s} q_s^{n_s - k_s} \tag{3.4}$$

where the subscripts denote the 2 populations.This gives a $\approx 2\%$ probability of producing these results based on a random guess for slow rotators and a negligible probability for fast rotators, suggesting a significant improvement. From the confusion matrix 3.7, there is a high level of success for fast rotators, but this is offset by the inaccuracy in predicting slow rotators, and the probabilities complementary in this case. This suggests that the algorithm was attempting to reproduce a detected ratio of fast to slow rotators blindly, and scoring inaccurately proportionally.

As can bee seen in 3.2, the When we look at the D/T dependence of the λ_{Re} value, we see a

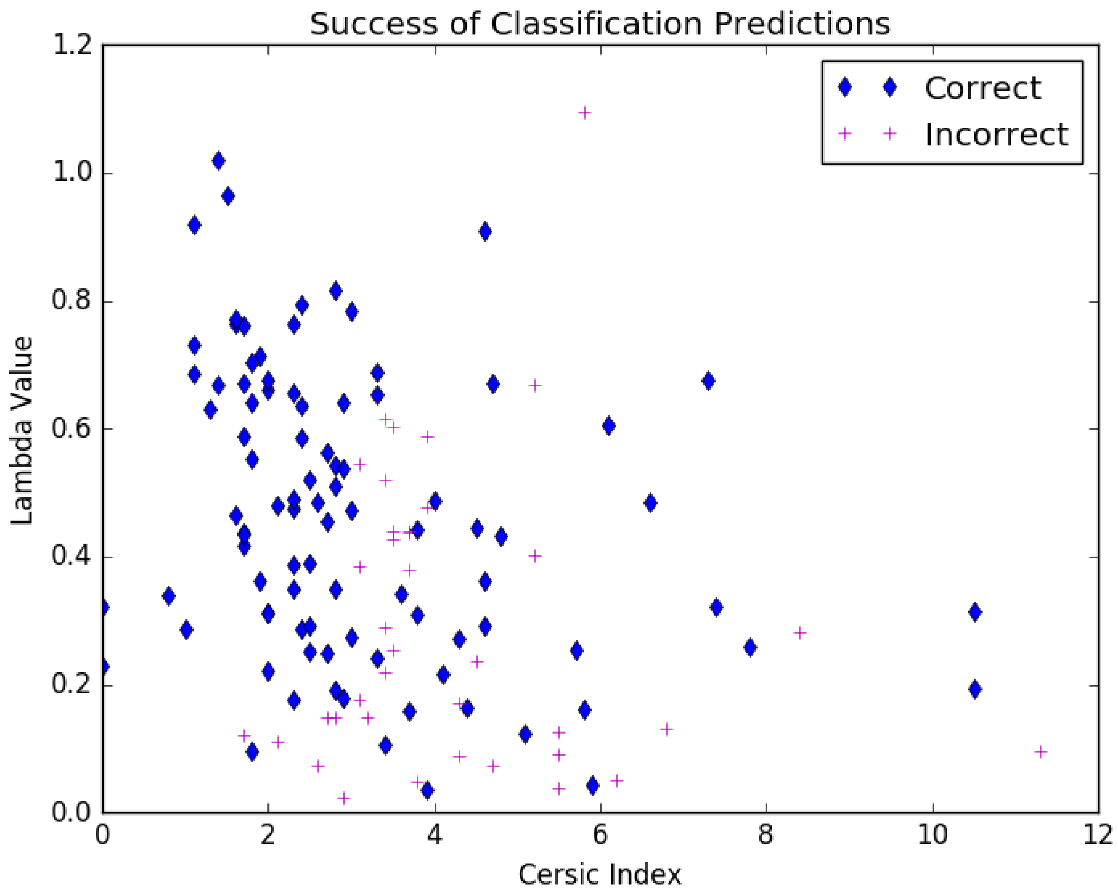


Figure 3.2: The success of the algorithm appears widely distributed and is not restricted to values above or below a threshold value of n.

more promising separation of the two populations, with a value of D/T ≈ 0.28 producing a low level of impurity. This parameter was identified as the most promising by [6]. The algorithm was then trained using D/T, improving the success rate to 81%. However, as can be seen in **??**, the success is again down to an over reliance on predicting fast rotators producing high level of success due to the different population sizes. This is emphasised when we evaluate the success of the decision tree classifier for those galaxies with no exponential disk component, where D/T$\lesssim$0.05, we find that the decision tree is still able to correctly classify the majority of galaxies. The success rate for just these galaxies is $\approx$77% which is more than that expected from the binomial distribution. However, the apparent success is undermined when we plot the

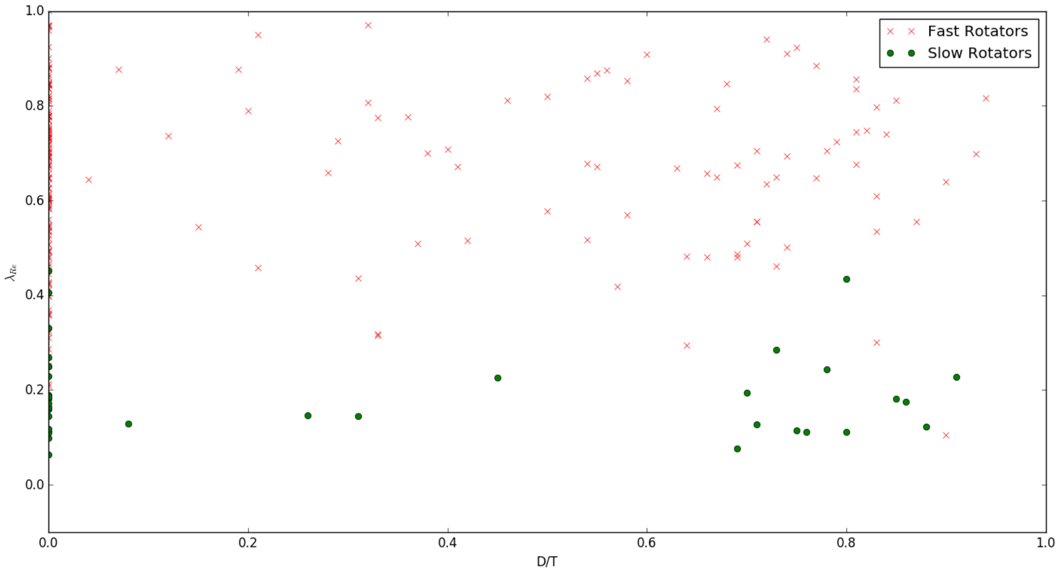


Figure 3.3: The separation between the two populations is more pronounced here. The galaxies with D/T = 0 have no exponential discs.

results for just these galaxies in 3.5. The algorithm benefited from the fact that only ≈10% of the galaxies with no disk in the test set were slow rotators. If the choice was made randomly, the success would be ≈15%, and so represents a significant improvement, but would hardly require such statistical learning methods to make the conclusion that if there is no disk it is most likely a fast rotator. The algorithm was then trained using both D/T and Seŕsic index of the single fit by passing the features as an $n \times 2$ matrix, resulting in a success rate of 75%, exceeding expectations based on random guess alone slightly, but is less than what was achieved using D/T alone. The plots 3.6 and 3.7 suggest that the algorithm is predicting slow rotators slightly more often rather than blindly assuming all galaxies are fast rotators, but it only successfully predicts a slow rotator once. These results suggest that the use of decision trees with parameters initialised to their defaults are not very promising. However, there are a large number of parameters, as outlined in the preceding sections, that can be tuned to maximise the efficiency of the classifier. SKlearn has the inbuilt function GridSearchCV that performs an exhaustive search over specified parameter values for an estimator. To avoid difficulties posed by a limited dataset size, the algorithm uses a k-fold cross-validation where:

> The original sample is randomly partitioned into k equal size subsamples. Of

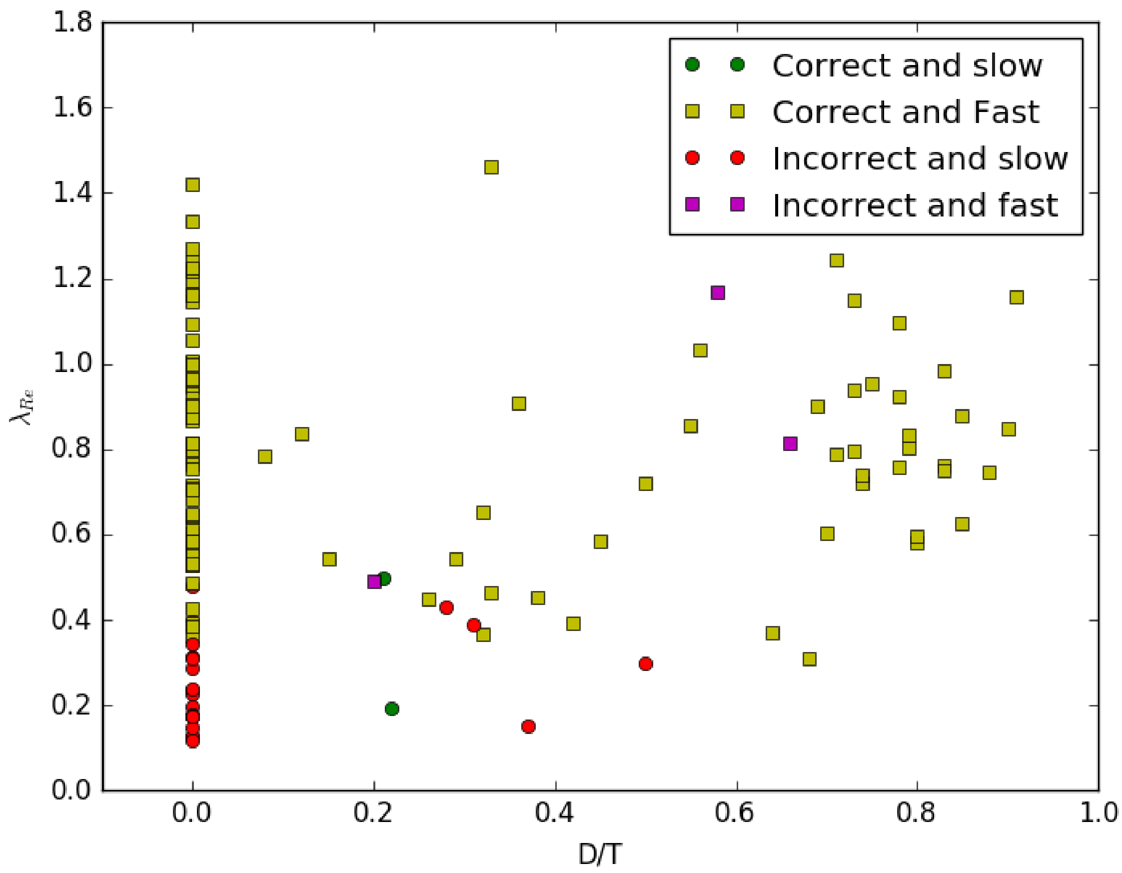


Figure 3.4: We see a high rate of success for fast rotators, but slow rotators are almost universally incorrect.

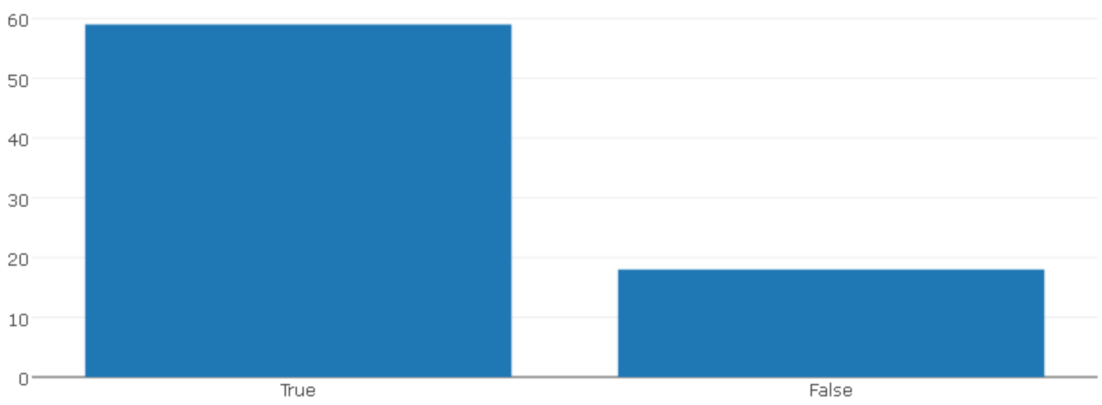


Figure 3.5: Prediction success for galaxies with D/T $\lesssim 0.05$.

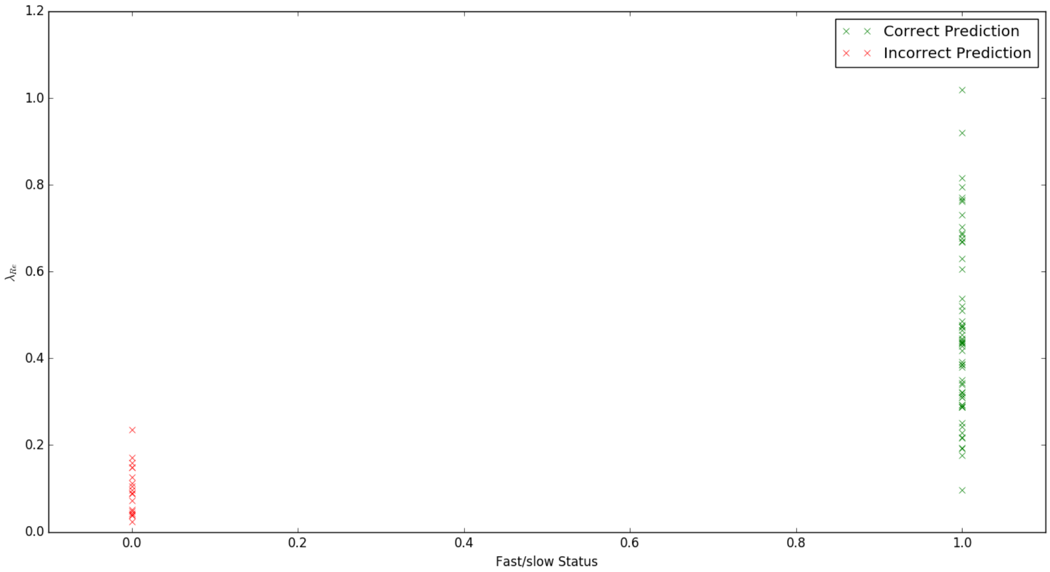


Figure 3.6: Investigating the results where D/T$\lesssim$0.05. We see that the algorithm predicts galaxies to be universally fast rotators.

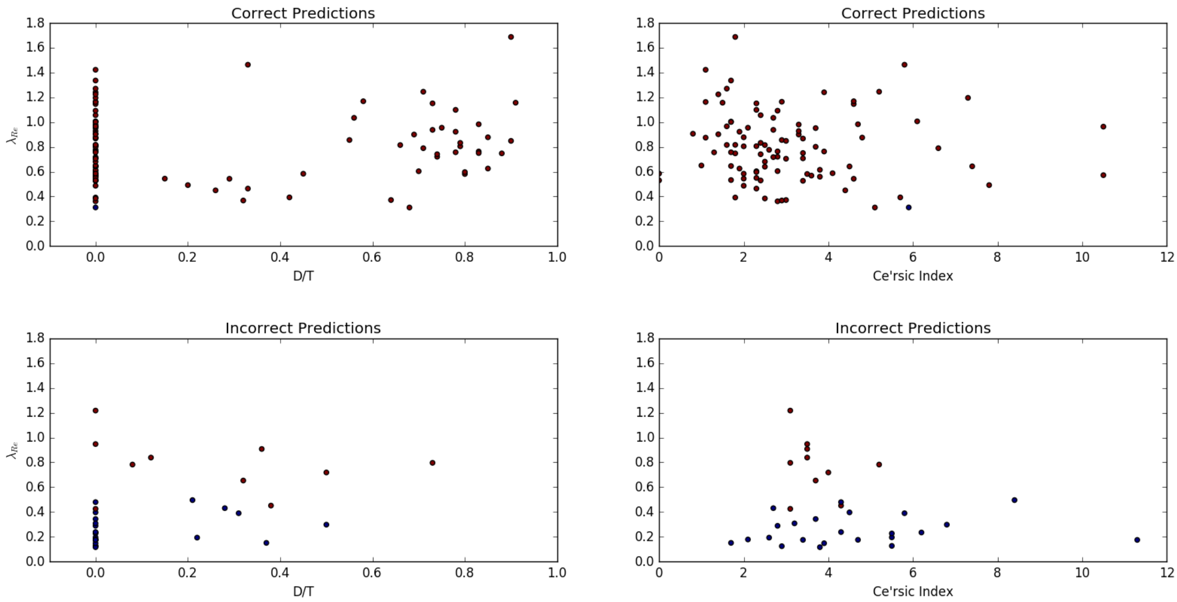


Figure 3.7: Evaluating the success of decision trees with 2 variables, seŕsic index and D/T. The markers are colour coded to be magenta for fast rotators and blue for slow rotators.

the k subsamples, a single subsample is retained as the validation data for testing the model, and the remaining k-1 subsamples are used as training data. The cross-validation process is then repeated k times (the folds), with each of the k subsamples used exactly once as the validation data. The k results from the folds can then be

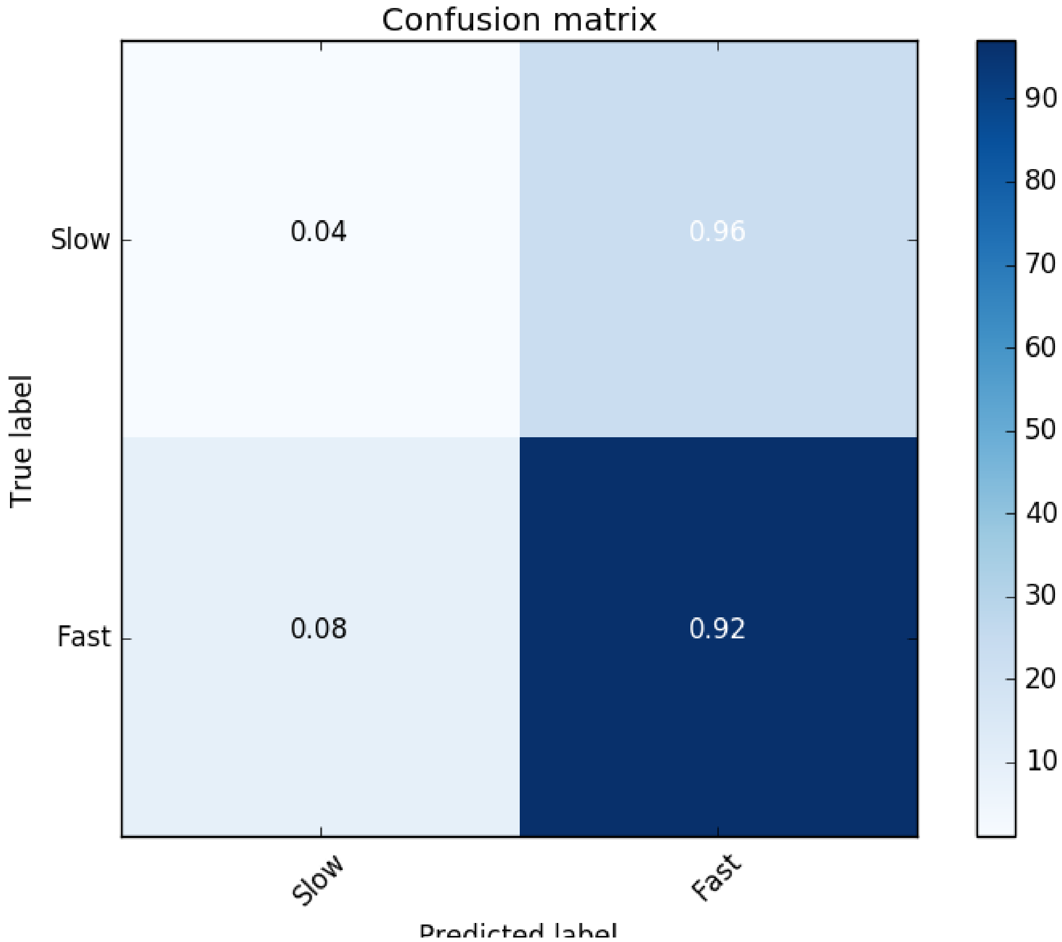


Figure 3.8: Confusion Matric for predictions based on D/T and Seŕsic Index.

averaged (or otherwise combined) to produce a single estimation. The advantage of this method is that all observations are used for both training and validation, and each observation is used for validation exactly once.[13]

This was performed using the possible options:

- criterion:['gini','entropy']
- splitter:['best','random']
- max_features:['sqrt','log2',None]
- max_depth:range(1,10)

In so doing, the success rate was increased to $\approx 93\%$, with the following options:

- criterion='entropy'
- splitter='best'
- max_features='sqrt'
- max_depth=4

In conclusion, decision trees offer promising results in identifying fast from slow rotating galaxies but requires some initial sorting of the data and careful model selection. The presence of a large number of galaxies with no disk component significantly skewed the results, and would best be initially filtered out since no new information is to be gained from their inclusion. Further study would focus on galaxies that have no disk component exclusively to see if better results can be achieved.

Bibliography

[1] E. Emsellem et al. "The SAURON project IX. A kinematic classification for early-type galaxies". In: *Monthly Notices of the Royal Astronomical Society* 379.2 (Jan. 2007), pp. 401–417. DOI: 10.1111/j.1365-2966.2007.11752.x.

[2] Houjun Mo, Frank Van den Bosch, and S. White. *Galaxy formation and evolution*. Cambridge University Press, 2010.

[3] Michele Cappellari et al. "The ATLAS 3D project I . A volume-limited sample of 260 nearby early-type galaxies : science goals and selection criteria". In: 836 (2011), pp. 813–836. DOI: 10.1111/j.1365-2966.2010.18174.x.

[4] Eric Emsellem et al. "The ATLAS 3D project III . A census of the stellar angular momentum within the effective radius of early-type galaxies: unveiling the distribution of fast and slow rotators". In: 912 (2011), pp. 888–912. DOI: 10.1111/j.1365-2966.2011.18496.x.

[5] Fabian Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2012), pp. 2825–2830. ISSN: 15324435. DOI: 10.1007/s13398-014-0173-7.2. arXiv: 1201.0490. URL: http://dl.acm.org/citation.cfm?id=2078195%7B%5C%%7D5Cnhttp://arxiv.org/abs/1201.0490.

[6] Davor Krajnović et al. "The ATLAS3D project XVII. Linking photometric and kinematic signatures of stellar discs in early-type galaxies". In: *Monthly Notices of the Royal Astronomical Society* 432.3 (2013), pp. 1768–1795. ISSN: 00358711. DOI: 10.1093/mnras/sts315. arXiv: 1210.8167.

[7] L. Cortese et al. "The SAMI Galaxy Survey: the link between angular momentum and optical morphology". In: *Monthly Notices of the Royal Astronomical Society* 000.August (Aug. 2016), stw1891. ISSN: 0035-8711. DOI: 10.1093/mnras/stw1891. arXiv: 1608.00291. URL: http://arxiv.org/abs/1608.00291%7B%5C%%7D5Cnhttp://mnras.oxfordjournals.org/lookup/doi/10.1093/mnras/stw1891.

[8] Bruce Simmons. *Binomial Probability Formula*. Feb. 2016. URL: http://www.mathwords.com/b/binomial_probability_formula.htm.

[9] Stephen Wilkins. *Extragalactic Astronomy Module Lectures*. 2017.

[10] SKDevelopers. *1.10. Decision Trees*. URL: http://scikit-learn.org/stable/modules/tree.html#tree.

[11] Unknown. *Overfitting Example*. https://kevinbinz.com/tag/overfitting/. Accessed: 2017-12-05.

[12] Unknown. *Scope of SDSS Survey*. http://www.808multimedia.com/winnt/kernel.htm. Accessed: 2017-02-05.

[13] Joaquin Vanschoren. *OpenML*. URL: https://www.openml.org/a/estimation-procedures/1.