Using Machine Learning to Classify Stars, Quasars, and Galaxies

Chinmay Ramamurthy, 11/29/2022

1. Abstract

In this project, we looked at data from stars, quasars, and galaxies from the sixteenth data release of the Sloan Digital Sky Survey Telescope. The aim of the project was to accurately and quickly classify these three types of objects using machine learning. We used three machine learning algorithms, namely logistic regression, multi-layer perceptron, and decision tree classifier. The classification was done based on measurements of the object's redshift and its u,g,r,i, and z light emissions. Logistic regression offered the least accurate results, with an overall accuracy of 94.83%, and a runtime of 28.2 seconds. Better in both aspects was the decision tree classifier, with an overall accuracy of 98.90%, and a runtime of 6.36 seconds. Several structures with different kinds of neuron-layer arrangements were tried for the MLP classifier. While all yielded around a 98% total accuracy, the more complex a structure was, or the more nodes and layers it had, the more time it took to run, and none even came close to the low amount of time taken by the decision tree classifier.

2. Introduction

The night sky is filled with billions of objects, including planets, stars, nebulae and galaxies, and each day, telescopes on land and in space take photos of these objects. The objects in these photos are classified both by observation with the naked eye as well as by the light spectrum these objects emit. One telescope which takes photos of these astronomical objects is the SDSS, or Sloan Digital Sky Survey telescope. The Sloan Digital Sky Survey's data has generated one of the most detailed three dimensional maps of the universe(1), and has detailed data on over three million astronomical objects. When telescopes need to catalog new objects found in space to use in future research, it's often helpful to know what kind of object it is. While many classification algorithms already exist for astronomical data, the accuracy of the classification can be continuously improved as newer machine learning algorithms come into place, and older ones are built upon. Here, we aimed to classify galaxies, stars and quasars with machine learning, using only the data provided in the SDSS table. This could create more accurate algorithms for scientists and researchers alike to study the telescope data, whether it be narrowing their search for a specific object/area, or observing more data about a particular object. For this project, we tested three different machine learning algorithms: decision tree classifier, logistic regression, and MLP neural network classifier. All three of these algorithms are considered supervised machine learning, meaning that the data we're working with is labeled (star vs. quasar vs. galaxy).

3. Background

For classifying stars, quasars, and galaxies from the SDSS dataset, Clarke et. al. and L. du Buisson et. al., use a random forest model, which is a method for classification that uses a group of multiple decision trees(2,3). Their accuracies were 97% and 90%, respectively. In our study we explored the efficiency of individual decision trees in classifying these objects.

L. du. Buisson et. al. also uses neural networks in his research, which inspired us to try that approach. Another researcher who uses neural networks in their approach is Carter Rhea. In his Medium article "Classification in Astronomy: Galaxies vs Quasar", he tries to solve a similar problem to ours: he wants to differentiate galaxies from quasars using SDSS bandpass data from AstroML(4). Our project, however, includes stars in addition to galaxies and quasars, to further expand on Rhea's work. He used neural networks as a potential classification model with a 99% accuracy for galaxies and 90% accuracy for quasars. His model had 2 layers with 100 nodes each. However, neural network models can almost always be improved upon due to the countless different structures it can have, as well as the lack of a guarantee that one structure will be better than another for a particular project. We attempt to build on this aforementioned work by exploring the models mentioned above, in addition to the logistic regression model, which is helpful to consider for any classification problem.

4. Dataset

Data:

In this work, we used the Sloan Digital Sky Survey's data. We used data from the 16th data release, which is the second most recent release as of October 2022. For this study, a table with information about 500,000 objects in space, consisting of 193,243 stars, 252,221 galaxies, and 54,536 quasars was used. These numbers are important, as they show the difference between the amount of stars, galaxies, and quasars that we can detect.

18 columns, including RA (longitude in the night sky), Dec (latitude in the night sky), field(section of the camera), redshift, and more were provided in the data. We focused on 8 features for

each object in this study. The first column is the “objid”, which is the Sloan Digital Sky Survey’s identification of the object. The second is “redshift”, which is an increase in the wavelength of the light emitted by an object. This can tell us how far an object is relative to Earth. The farther an object is from Earth, the greater the redshift. If a telescope detects a star, that star is most likely in or around our own galaxy, the Milky Way. This means that the star would have a smaller value for redshift than a galaxy or quasar. The third column we used was the “class” of the object, which in the data release was one of GALAXY, STAR, or QSO(Quasar). We renamed these to 0,1, and 2, respectively, for convenience in the code. The remaining five measures used were the object’s bandpass measurements, namely the u, g, r, i, and z. These are measurements of light at different parts of the electromagnetic spectrum. U is ultraviolet, G is green, R is red, I is Infrared, and Z is anything beyond. Each object emits a certain amount of light in each of these spectral sections, and these light emissions could vary based on what type of object it is.

The 8 parameters mentioned above were used to train the model, and all the other parameters were dropped. Most of the values for u, g, r, i, and z were between 10 and 20. However, there were some entries showing values like “-1000”. This could be due to an error in measurement or corrupt memory. It could also be due to the telescope not being able to properly measure these values due to distance or something obstructing the view. There were only around 8 of these, so we changed the value to 1, to not raise an error during division (if we had set it to zero). In addition, we added a parameter called “ratio”. This parameter is the ratio of each object’s value of u to g to r to i to z.

Visualization:

To visualize the data, we used the pandas library’s .DataFrame structure. Histograms and boxplots, created using built in python functions, matplotlib, and numpy, were helpful in looking at trends in the data and developing strategies for analysis. The average values for the bandpass attributes, u, g, r, i, and z, as well as redshift, differed for each type of object in the dataset, as shown by the histograms, mentioned below.

We created histograms for each bandpass of light in u, g, r, i, and z, and displayed the distribution for each type of astronomical object(Figure 1). We noticed that for each graph, the general shape of the histogram for each type of object remained the same, although at a different scale due to the different quantities for each object provided by the SDSS dataset. However, while the shape remained the same, the peaks of each histogram were shifted slightly, showing the difference between the objects, and suggesting that the bandpass parameters would be useful in classifying the data.

We also looked at histograms and boxplots for the redshift(Figure 2 and 3). As we discussed earlier, the redshift values for the stars were much lower, most being from -0.002 to 0.002. This likely means that the stars are actually getting closer to us. The histograms for the galaxies showed almost all of the galaxies having a redshift value from 0.05 to 0.15. Meanwhile, the redshift values for quasars showed the quasars almost evenly distributed from 0.05 to 0.75. This could make it challenging to tell apart quasars with a redshift value near 0.1, as that is where most of the galaxies' redshifts lie.

5. Methodology/Models

The data was split randomly into training and testing sets, with the training set having 400,000 values and the testing set having 100,000 values. This 80:20 ratio is a common split used in many machine learning algorithms, as it uses most of the data to train with, preparing the model for most cases. However, it still leaves sufficient data to test the algorithm with, making sure the results are accurate. We used functions from the scikit-learn python package to fit the model with the training data and analyze our results.

For this project, we used 3 different kinds of models, namely the logistic regression model, the MLP classifier, and the decision tree classifier to classify the data into galaxies, stars, and quasars. We tested each model with the eight parameters above, excluding objid, as it is just the object label. In addition, we did not use the ratio parameter for all of the models, rather just using it for one test of the MLP classifier.

The logistic regression model creates a graph, with the attributes of the objects as the x-axis and the probability of it being one or the other(say, a galaxy or a star), on the y axis. If there are multiple attributes, the x axis has multiple dimensions. It maps a logistic sigmoid function, which is a function that starts above an asymptote at 0% probability, and increases rapidly until it reaches 50%, then has an inflection point, continuing to increase, but slowing down until it has an asymptote at 100% (Figure 7). The model uses the output data of the sigmoid function to predict the probability of the object's classification. Here we use 6 attributes on the x-axis, namely u,g,r,i,z, and redshift. To train and test our data, we used the LogisticRegression function from the sklearn package's linear model section. We set the random state to zero to make our results more reproducible, as we had already randomized the data before we ran the algorithm.

The decision tree classifier is a model which, using a set of rules, generates splits in the training dataset repetitively, and then finds the split which makes the most progress toward classifying the data.

We can visualize this model by thinking about a typical tree in graph theory. All the data starts at the root node, and each subsequent node represents a decision. Based on the outcome of the decision, the object in the testing data gets sent to one of that node's children, and the process continues until the branch contains all data from one category or if the maximum tree depth is reached. We used the sklearn.tree.DecisionTreeClassifier package to generate the Decision tree classifier model.

The multi layer perceptron is a type of neural network that functions in a similar way to our own brain, which is built up of many neurons which are organized in layers. Each layer takes in input from a previous layer and passes on output to other layers until it is consolidated into a final decision on the classification. For the MLP classifier, since it was the main focus of this project, we tried a few different structures at random, as there is usually no clear way to decide which structure is best for a particular classification problem. We tried a single layer with 10 nodes, 2 layers with 8 nodes each, 3 layers with 15 nodes each, and 5 layers with 8 nodes each. In addition, we tried models both with and without the ratio parameter, which is the ratio of all of the bandpass values(u,g,r,i,z).

6. Results and Discussion

Logistic Regression

Out of the three models we tried, logistic regression offered the least promising results, although there was still an overall accuracy of 94.83%, meaning that 94.83 percent of the objects were classified in accordance with the SDSS's classification. Something to note, however, is that the percentage of stars which were classified by logistic regression in accordance with SDSS was 96.73%, while the percentages of galaxies and quasars that were classified correctly were 93.97% and 95.29%, respectively(Figure 4).This difference could be due to the fact that quasars and galaxies can have similar values for redshift and could be confused by the algorithm for one another.

Logistic regression takes a relatively long time to run, and is the least accurate out of the three algorithms used in this study. Other studies, such as Ethiraj et. al, have reported similar accuracies of about 95% using logistic regression(5). They also tested models such as the extra trees classifier, random forest classifier, K nearest neighbors, and even decision tree, which we also tested. In their study, however, logistic regression's accuracy was on par with or greater than 11 of the 13 other models they tested, including decision tree, which outperformed logistic regression in our study by around 4%. This difference could be because of the 7type of data used. Our studies use only tabular data while their study uses images as well as tabular data.

MLP Classifier

The MLP classifier, being a neural network, had the most diverse structures to explore before a conclusion could be reached. The overall accuracies for the models with 1 layer of 10 nodes, 2 layers with 8 nodes, 3 layers with 15 nodes and 5 layers with 8 nodes were 98.82%, 98.64%, 98.82% and 98.78% respectively (Figure 5). We observed that using complicated structures did not improve the accuracies compared to the 1 layer and 10 nodes. The differences, if any, were not more than 0.3%. On the other hand, using complicated structures definitely resulted in increased runtimes(1 min 53 sec, 3 min 7 sec and 2 min 13 sec respectively) compared to the 1 layer and 10 nodes (1 min 37 sec). The original model with 1 layer and 10 nodes identified stars with 99.92%, quasars with 95.37% and galaxies with 98.81% accuracy. In addition, we reran the model with 1 layer and 10 nodes, but added in the previously mentioned ratio parameter, which had an accuracy of 98.92% (Figure). Stars were identified with 99.95% accuracy, galaxies with 98.59% accuracy and quasars with 96.80%. The addition of the "ratio" to the model consistently generated better results, but only improved the accuracy by 0.1%, and took more time than the original model, at 2 min 3 sec. As compared to logistic regression, the MLP classifier was more accurate, but took nearly 4.4 times as long as the logistic regression. While the MLP classifier offered much more promising results than logistic regression, any tweak or change to the algorithm made the algorithm take a longer time, with minimal, if any, change to the accuracy.

Decision Tree Classifier

The decision tree classifier and the MLP classifier offered similar results in terms of accuracy. The decision tree classifier's overall accuracy was 98.98%. It identified stars with a 99.75% accuracy, only misclassifying 95 stars. It was also good at identifying galaxies with 98.93% accuracy. However the accuracy of detecting quasars was considerably lower at 96.47%(Figure 6). Quasars were most likely misidentified as galaxies, as a quasar is a burst that can come from the center of a galaxy, meaning that the quasars in the dataset would have the same redshifts on average as the galaxies. If there were data about the shape of the object, it is possible that fewer quasars would be misidentified. We believe that the decision tree classifier is the best overall algorithm to use as it takes the least time to run at only 6.36 seconds. It performs at least as well as the MLP classifier. In the training data, the decision tree had 100% accuracy. This is likely due to the way that the algorithm is built, creating "splits" in the data which are based on the training set data, meaning that it would match exactly. This brings the possible question of overfitting in the data. However, the high accuracy of the algorithm shows that this is not a major issue here. See the decisions made by the tree (https://tinyurl.com/DecisionTreeChart)(6)

Overall, the decision tree algorithm performed the best in both the measures of success that we used, namely time and accuracy. In terms of time taken to run, the decision tree was undoubtedly the best. However, the neural network structure with 1 layer, 10 nodes, and the "ratio" parameter did surpass the decision tree by a small amount in accuracy for the testing set. This structure, however, took 1 minute and 57 seconds longer than the decision tree. This amount of time may not seem that large, but if we wanted to run this algorithm on a dataset 100 times the size of the one here, this would be a major concern. This makes the MLP classifier far less practical to use, especially when small differences in percentage accuracy like 0.02% will not affect a study(Tables 1 and 2).

7. Conclusion

In this project, we aimed to classify galaxies, stars, and quasars from the SDSS DR-16 dataset with an accuracy of 93% or more using machine learning. We achieved this goal by using logistic regression, decision tree classifier, and MLP neural network classifier. These models were chosen based on what worked and didn't work for other researchers as described in the background section above. The decision tree classifier performed with a high accuracy of 98.90%. This is likely due to the fact that a decision tree can accurately split the data based on common trends found in all stars, galaxies, or quasars. The MLP classifier produced a similar accuracy to the decision tree, always staying within 0.5% of the decision tree model's results. However, the time taken to run the MLP classifier, especially using models with more nodes and layers, was 1 to 3 times more than that taken by the decision tree. This shows that it is more beneficial and generally useful to use a quick algorithm as opposed to a slow one which may or may not give better results. Overall, these algorithms proved to be very accurate in their classifications, most of which matched the SDSS classifications over 97%. This can help better classify fresh data from a telescope, and confirm classifications already made.

To further improve this research, several extensions of the project can be made. First, it is possible to experiment with other different kinds of machine learning algorithms, including, but not limited to, linear regression, k-nearest neighbors(KNN), random forest classifiers, and types of similarity learning. These algorithms may be able to classify the objects more accurately. In addition, different kinds of structures for neural networks can also be experimented, as there are a limitless number to try. However, this may not improve on our previous neural network results, as no change in structure during our testing yielded a result over 1% better. Another issue common in machine learning is that of overfitting. Overfitting is when the model is too finely tuned to objects in the training dataset and applies generally worse decisions to classifying the testing data. The main drawback of this is when there are outliers in the data and the model fits the decisions too closely to the outlier. To solve this, it's possible to remove certain values as outliers, and train the model with the rest of the data. While this makes us lose valuable information, it could possibly help increase accuracy. The next steps mentioned above are just a few of the possible directions that this project could be taken. Truly, this project can be as diverse as the objects it classifies.

8. Acknowledgements

We would like to acknowledge Ms. Amanda Wang for her direction, mentorship, and editorial help in writing this manuscript. We would also like to thank Inspirit AI's 1:1 Research Mentorship Program for providing the training, structure, and basis for this project.

9. References

1. https://www.sdss.org
2. L. du Buisson, N. Sivanandam, Bruce A. Bassett, M. Smith, Machine learning classification of SDSS transient survey images, Monthly Notices of the Royal Astronomical Society, Volume 454, Issue 2, 01 December 2015, Pages 2026–2038. https://academic.oup.com/mnras/article/454/2/2026/1051683
3. A. O. Clarke, A. M. M. Scaife, R. Greenhalgh and V. Griguta, Identifying galaxies, quasars, and stars with machine learning: A new catalogue of classifications for 111 million SDSS sources without spectra A&A, Volume 639, July 2020 https://www.aanda.org/articles/aa/full_html/2020/07/aa36770-19/aa36770-19.html
4. https://medium.com/swlh/classification-in-astronomy-galaxies-vs-quasars-ff3069dcfbe3
5. Ethiraj S, Bolla BK, Classification of Quasars, Galaxies and Stars in the Mapping of the Universe Multi-modal Deep Learning, arXiv:2205.10745
6. Decision Tree Decision Chart: https://docs.google.com/document/d/1vzSWCsMJ3YSbin7gmB3c-ah95Oqg6fz5bBTwG4JDT5A/edit

10. Figures and legends

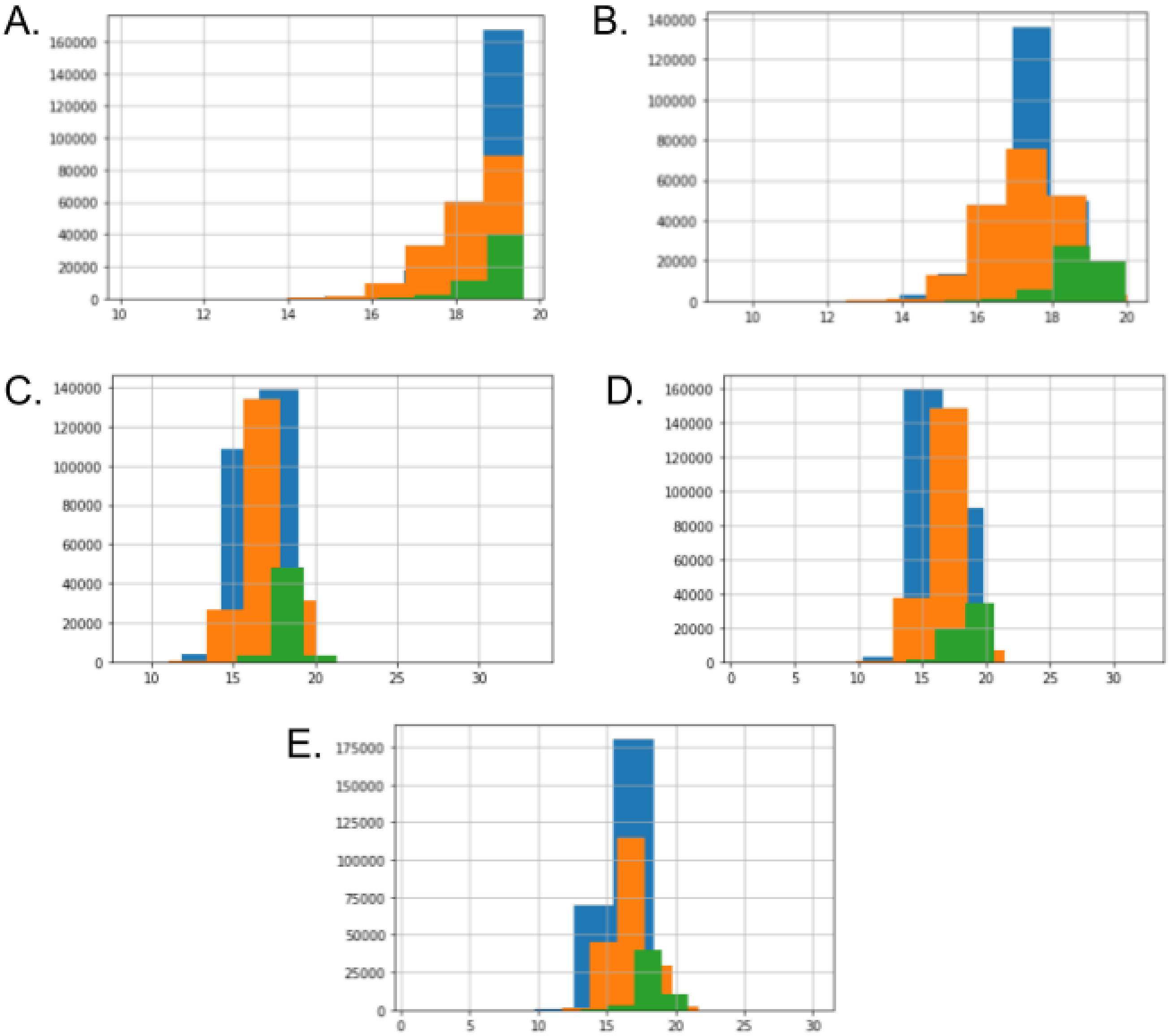

Figure 1: Histograms of the u,g,r,i,and z bandpass emissions(Blue = Stars, Orange = Galaxies, Green = Quasars) (A) "u",(B) "g", (C) "r", (D) "i" and (E) "z"

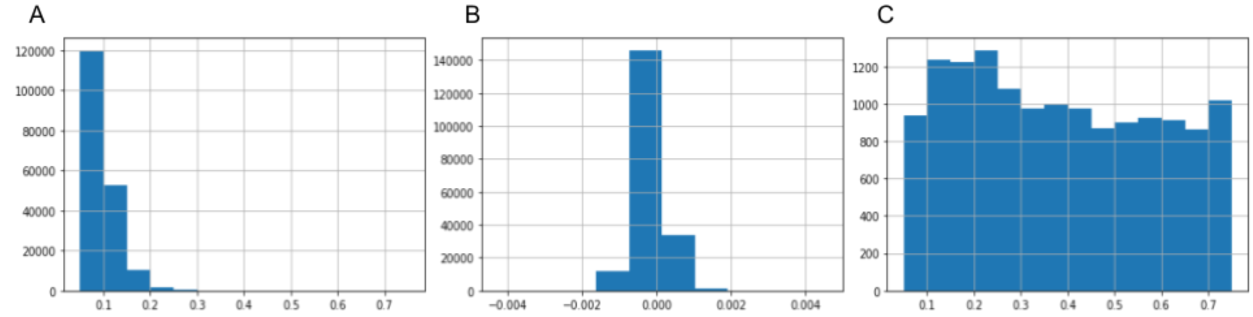

Figure 2: Histograms with the redshift distributions for the three types of objects (A) Galaxies, (B) Stars and (C) Quasars

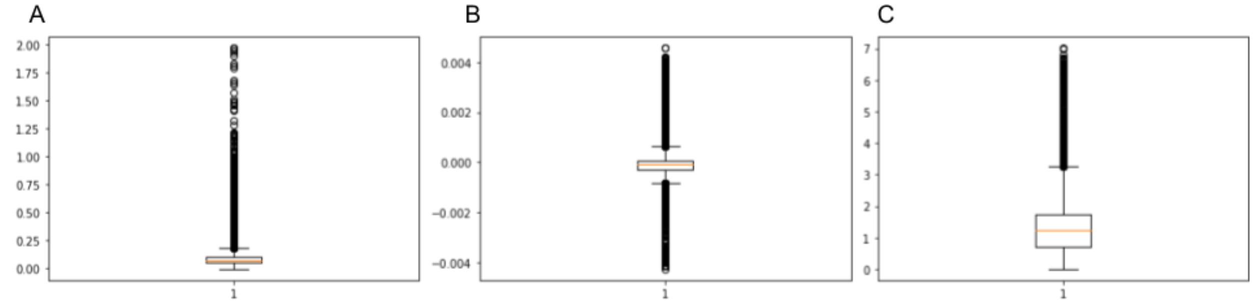

Figure 3: Boxplots with the redshift distributions for the three types of objects(separate points seen are outliers) (A) Galaxies, (B) Stars and (C) Quasars

Model	**Overall Accuracy**	**Galaxy Accuracy**	**Star Accuracy**	**Quasar Accuracy**
Logistic Regression	95.09%	94.03%	96.65%	94.44%
MLP Classifier - 1layer 10nodes	98.80%	98.86%	99.89%	94.69%
MLP Classifier - 2layers 8nodes	98.40%	98.26%	99.99%	93.42%
MLP Classifier - 3layers 15nodes	98.90%	98.50%	99.90%	97.21%
MLP Classifier - 5layers 8nodes	98.90%	98.81%	99.86%	95.93%
MLP Classifier - 1layer 10nodes with "ratio"	98.87%	98.62%	99.93%	96.26%

Decision Tree	100%	100%	100%	100%

Table 1: Training accuracies for all the models

Model	Overall Accuracy	Galaxy Accuracy	Star Accuracy	Quasar Accuracy
Logistic Regression	94.83%	93.97%	96.73%	95.29%
MLP Classifier - 1layer 10nodes	98.82%	98.81%	99.92%	95.37%
MLP Classifier - 2layers 8nodes	98.64%	98.28%	99.99%	94.31%
MLP Classifier - 3layers 15nodes	98.82%	98.45%	99.91%	97.59%
MLP Classifier - 5layers 8nodes	98.78%	98.79%	99.87%	96.57%
MLP Classifier - 1layer 10nodes with "ratio"	98.92%	98.59%	99.95%	96.80%
Decision Tree	98.90%	98.90%	99.79%	95.77%

Table 2: Testing accuracies of all the models

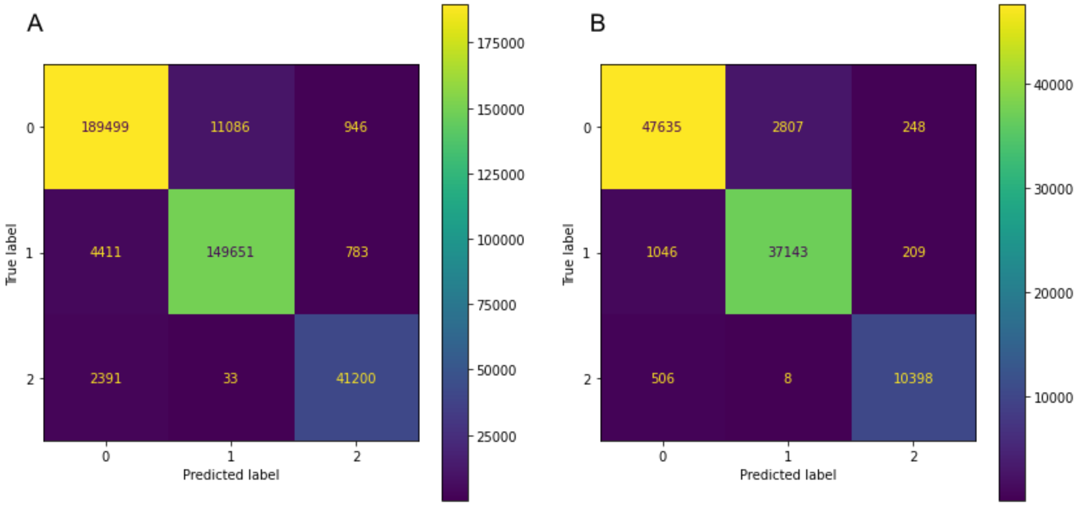

Figure 4: Confusion matrix of Logistic regression (A)Training set and (B) Test set. The predicted label is the model's classification while the true label is the SDSS's classification for the object. 0 represents galaxies, 1 represents stars, and 2 represents quasars.

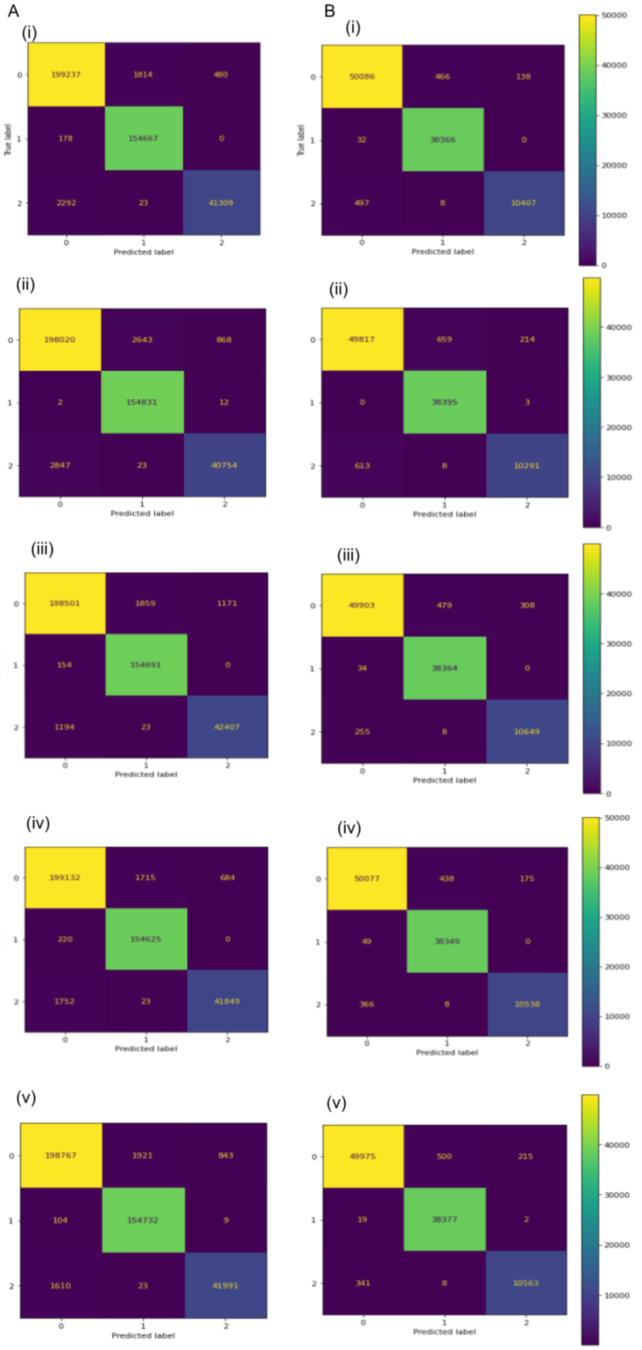

Figure 5: Confusion matrix of MLP Classifier (A) Training set and (B) Test set. (i) 1 layer and 10 nodes, (ii) 2 layers and 8 nodes, (iii) 3 layers and 15 nodes, (iv) 5 layers and 8 nodes, (v) 1 layer and 10 nodes with "ratio". The predicted label is the model's classification while the true label is the SDSS's classification for the object. 0 represents galaxies, 1 represents stars, and 2 represents quasars.

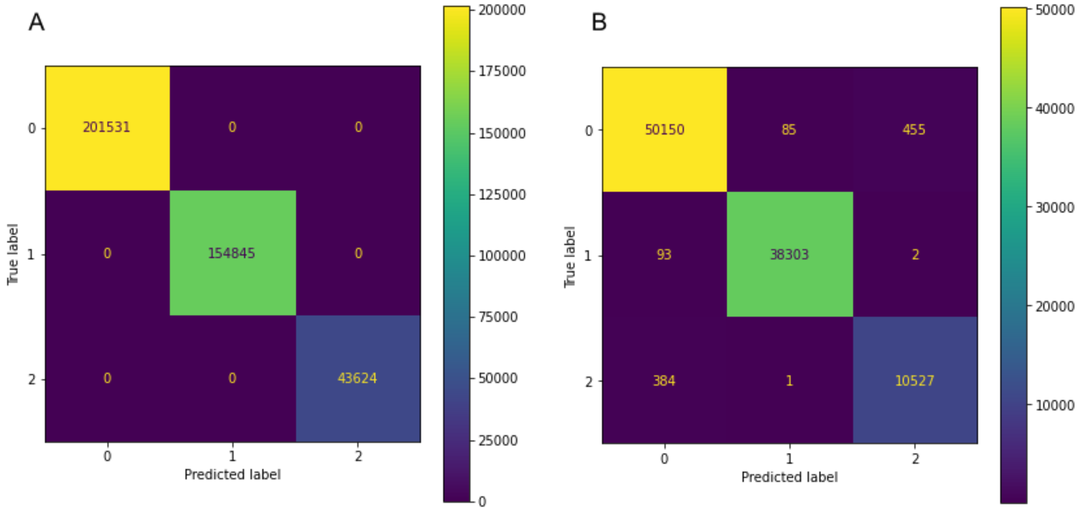

Figure 6: Confusion matrix of Decision Tree Classifier (A) Training set and (B) Test set.The predicted label is the model's classification while the true label is the SDSS's classification for the object. 0 represents galaxies, 1 represents stars, and 2 represents quasars.

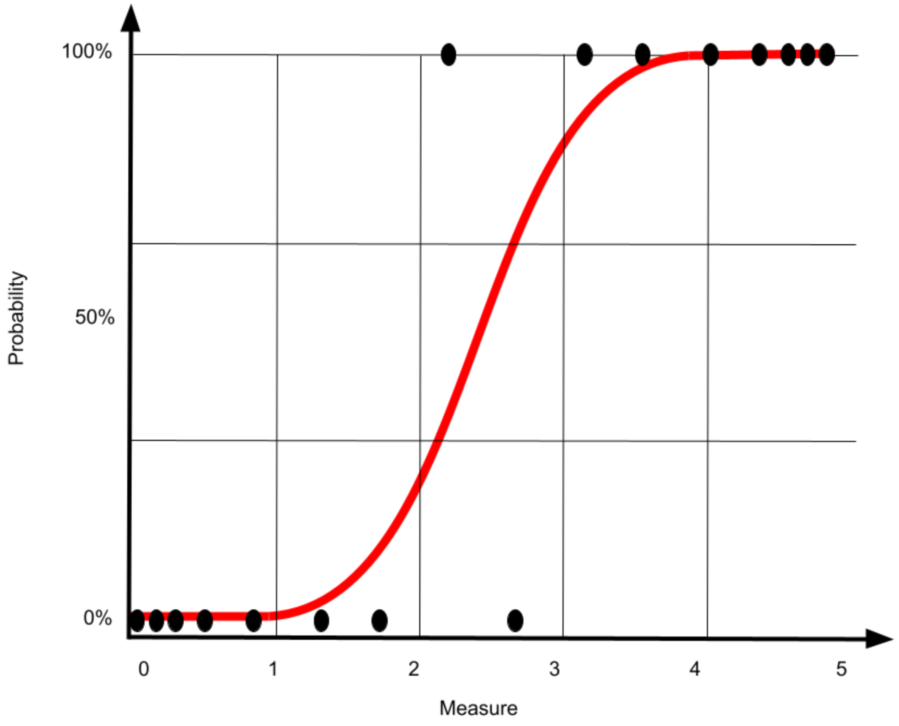

Figure 7: Example logistic regression sigmoid graph with one attribute.