DS-252: Classification Report

Michael Kachanyuk, Finn Browning, Evan Works.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

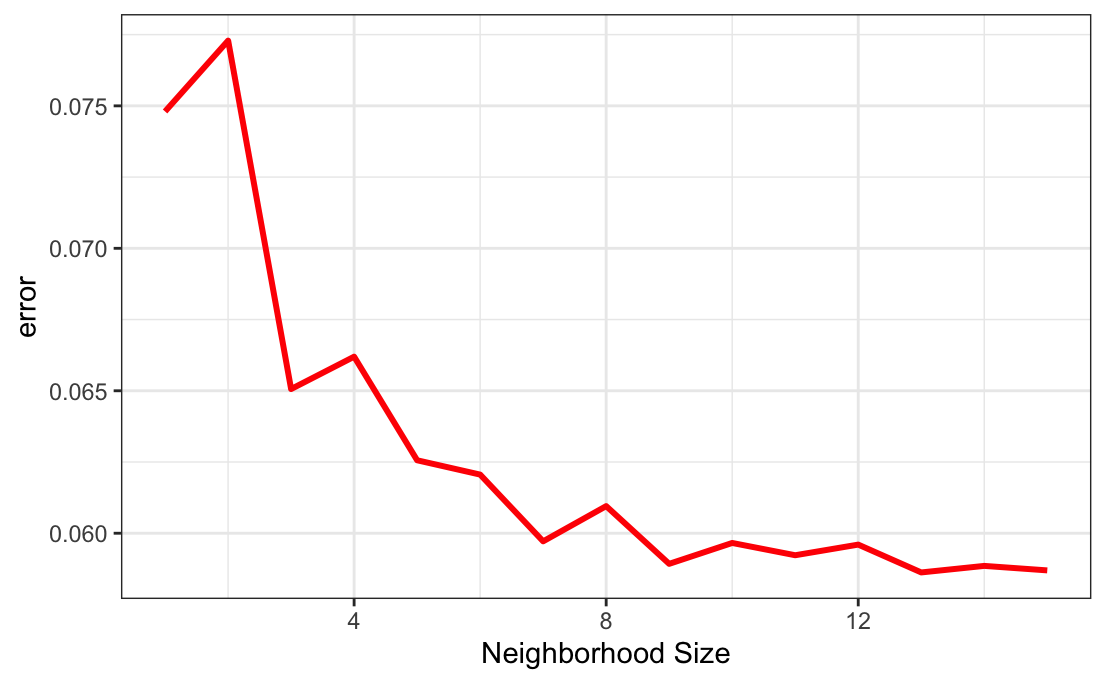
Introduction

In this project, we analyzed three types of celestial objects: stars, quasars, and galaxies. The data was collected from the European Space Agency (ESA) mission Gaia and processed by the Gaia Data Processing and Analysis Consortium (DPAC). The DPAC has been funded by national institutions, particularly the institutions participating in the Gaia Multilateral Agreement. Three data sets were collected from the DPAC; one about each celestial object we are analyzing, and were combined into a single data set we call space. The goal of the project is to create training and testing data sets to develop classification models that will be used to make predictions with new data. We will use K-nearest neighbors (KNN), classification trees, bagging, linear/multilinear regression, linear discriminant analysis, and quadratic discriminant analysis models. Additionally, we will be comparing these models to historic models from past work.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

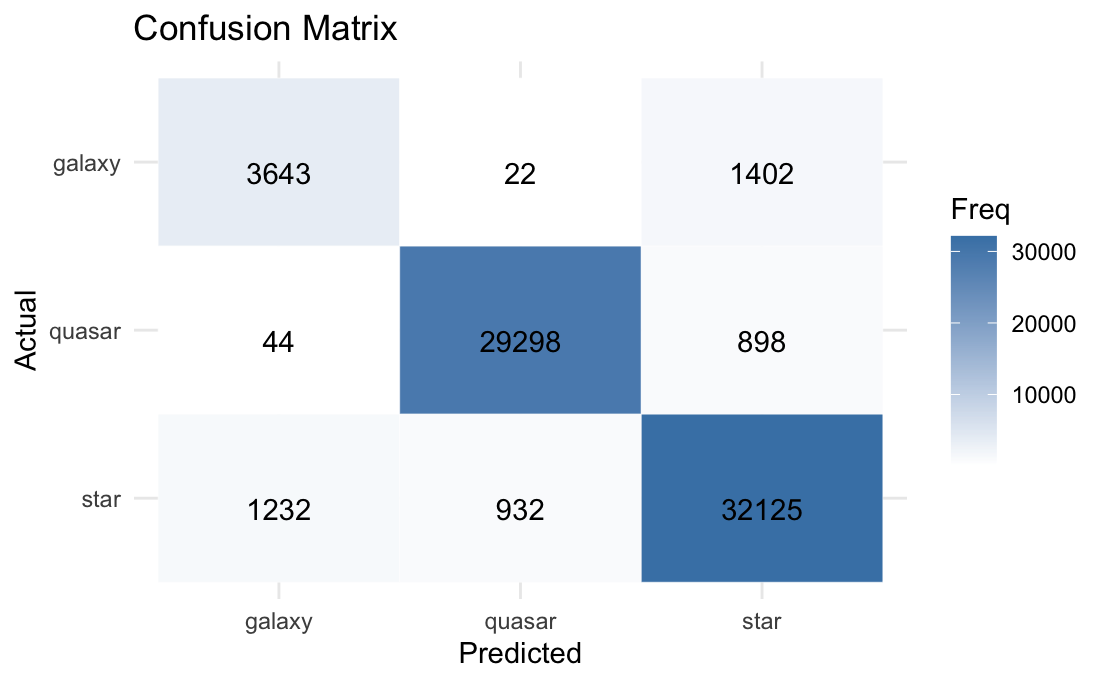
KNN

The first model we will look at is KNN. This model essentially works by looking at data points and classifying a group of points based on the other points surrounding it. With a lower k-value you are looking at a smaller group of “neighbors” which can lead to overfitting whereas a higher k-value can lead to greater bias. Before analyzing our data with KNN, we first need to split our data into training and testing groups with a 60-40 split. Next we will use the function “knn” from the library “class,” specifying our training and testing sets, as well as our classification variable within our training set. This provides us with a model that has an accuracy rating of about 93.49% and an error rating of approximately 6.51%. Now it should be known that these values were obtained with a k-value of 3. Since too low of k-values can result in overfitting, we should now make a “grid search” to make sure our k-value is not overfitting our data. Through this process we get the following graphic:



By looking at this graph we decided that our initial k-value of 3 was a good fit for our model. Now that we are confident with our k-value we can make a confusion matrix to gain a better sense of our accuracy and error ratings. This leads us to the following matrix which to clarify has the same accuracy and error rating as our KNN model.

*KNN Confusion Matrix Below:*

****

Looking at this visualization, we can see that this model does not mistake quasar’s for galaxies very often or vice versa. This means a bulk of the error comes from galaxies and stars getting confused for each other and to a lesser extent quasars and stars. Overall KNN was reasonably accurate at modeling our data. However, there is definitely room for improvement because we still had an error rating around 6.51%, but we will check other models and use this rating for future reference.

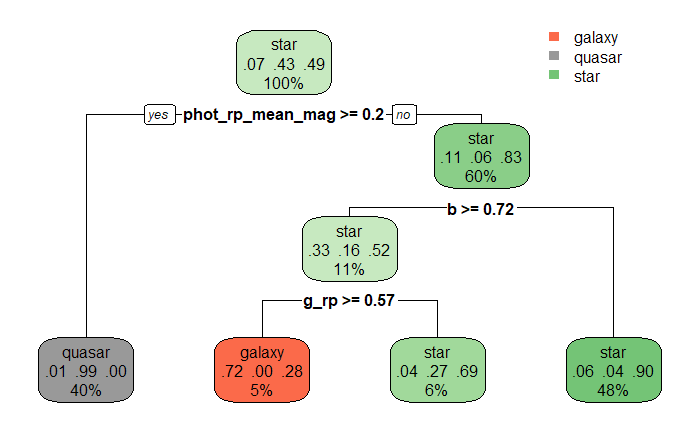
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tree and Bagging

After completing our KNN model, we decide to move on to a decision tree. Decision trees work off of a if-then basis to model potential outcomes and consequences of a sequence of decisions, hence the name. First, the algorithm starts by splitting at the root node, which is the entire dataset in this case, and splits the tree into different sub-nodes based on certain attributes of the data. It does this by trying to find the best outcome that maintains the most homogeneity of data. This splitting occurs until either a certain predefined criteria is met or the data can no longer be split further. After the tree is built, new data can be fed into the model and it will be either classified or predicted based on what path the data takes through the tree.

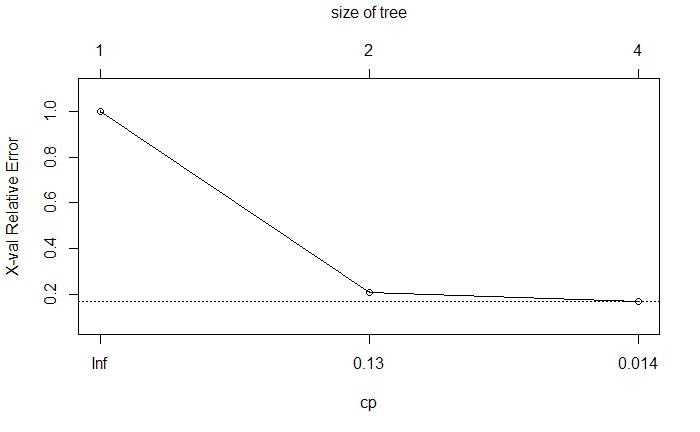
Moving on to creating our decision tree, we will use the training dataset that we split earlier for our KNN model. We will use “rpart” from the “rpart” library and “rpart.plot” from the “rpart.plot” library to one, make our trees and two, visualize them to easily discern how our model splits and with which variables the tree decides to split. With the tree made, we can see that the first variable that the tree decides to split at is the “phot\_rp\_mean\_mag” where it splits if the value is greater than or equal to 0.2. The variable refers to the mean magnitude in the integrated RP band, RP band being the red photometer on the GAIA mission, which is part of the larger spectrophotometer system. All this to say that the tree first splits looking at the wavelengths of light that are within 610 and 1050 nanometers. Our tree then shows that if this magnitude is greater than or equal to 0.2 then it decides that the body is a quasar with 40% of our dataset falling into this category. The next split occurs with the “b” variable which is the galactic latitude, this is relative to the Milky Way Galaxy. The final split of our tree occurs with the “g\_rp” variable which is the “phot\_g\_mean\_mag” - “phot\_rp\_mean\_mag”. “phot\_g\_mean\_mag” is similar to our “phot\_rp\_mean\_mag”, however, it is with the G Band, which deals with photometric data much more similar to our Sun. This makes sense since it has to do with splitting galaxies from stars.

*Tree Below:*



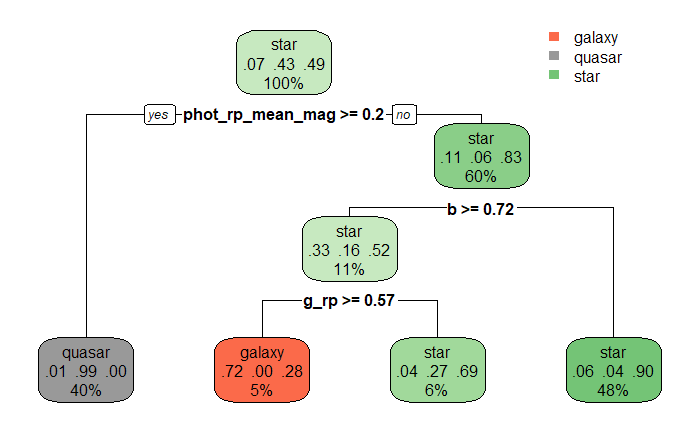
Now that we have created a decision tree, we want to move to pruning this tree which will help prevent overfitting. Below is our plotted cp of our above tree, as we can see there is very little change, which makes sense since our tree is so small to begin with.

*Diagram Below:*



Now that we have a plotted cp value we can modify our code to augment our tree with this new cp value. As shown below, however, there was no change in the tree whatsoever which is a little surprising that we didn’t see any change in the values either.

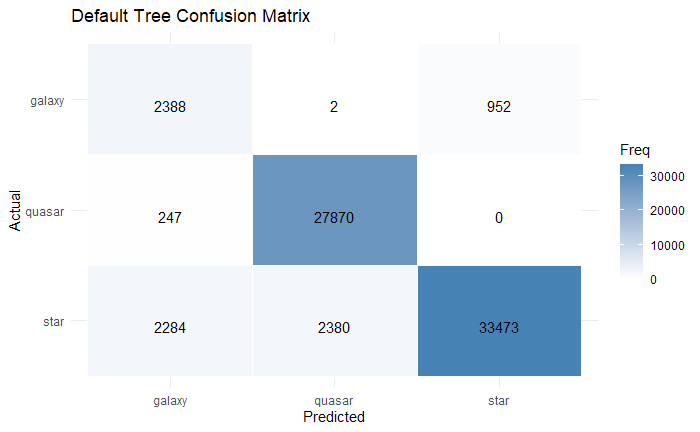
*Decision Tree Below:*

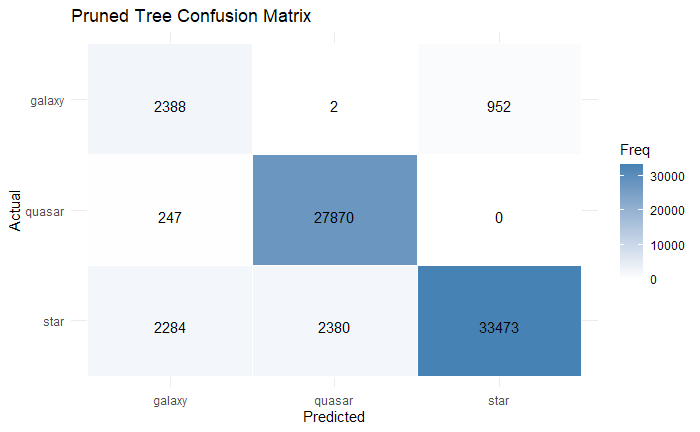


After successfully pruning the tree, even though there was nothing that was pruned, let’s move on to confusion matrices for these two trees to determine their effectiveness in these classifications.

To start us off, we will take our testing portion of the dataset that we set aside earlier to get a sense of how well our decision works. As expected, the two confusion matrices are the exact same. Looking at the matrix itself, it shows that our model was very good with classifying quasars and stars but somewhat struggled with galaxies. This is somewhat odd since when looking at galaxies and stars, there are clear defining features that separate the two from one another. The issue that seems to arise is that the tree looks primarily at spectrophotometric data, which makes it more difficult to discern a galaxy from a star.

*Default and Pruned Tree Confusion Matrix Below:*

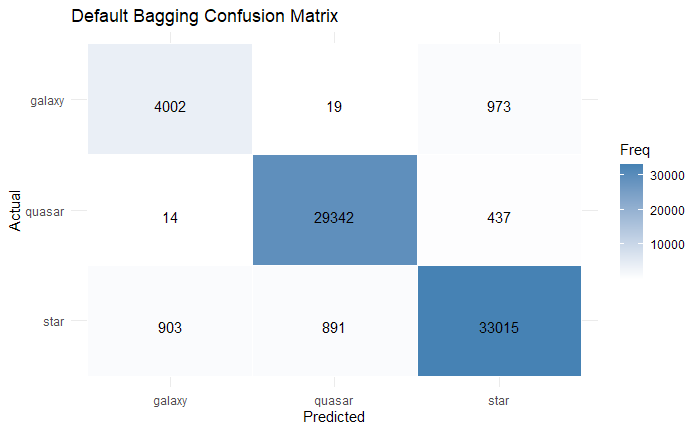




The next classification technique we will employ is bagging, also known as Bootstrap Aggregating, which is a method that trains multiple models on different subsets of the training data. Firstly, we use the same training and testing dataset splits we did earlier, next we will use the bagging function to generate our model and set it up into a confusion matrix just like we did with the decision trees so that we can easily visualize the effectiveness of this method.

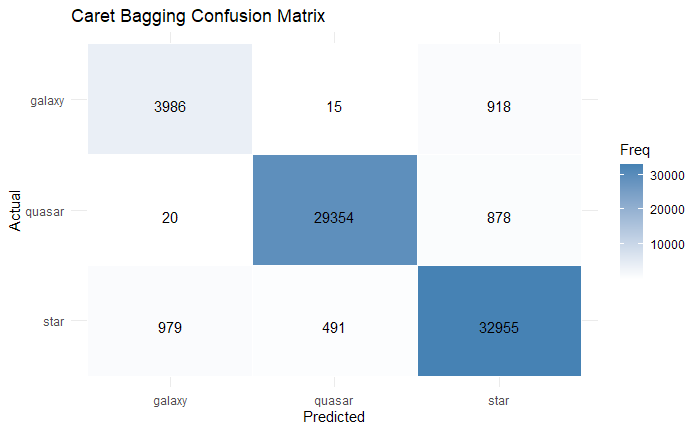
With the default bagging technique we can see that it has a rather effective classification probability, especially with classifying galaxies when compared to our tree models. This model has a 95.34% accuracy rate when classifying with our testing data.

*Bagging Confusion Matrix Below:*



Now let’s try it with the “caret” package and see if there is any improvement with our accuracy percentage. Below we can see that the values themselves have changed slightly but there doesn’t appear to be any serious change in the values. The accuracy percentage for caret bagging is 95.25% so it’s slightly worse than our original bagging classification but not by much at all.

*Caret Confusion Matrix Below:*



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Simple Logistic and Multiple Logistic Regression

Our final models will be the simple logistic and multiple logistic regression models. These models estimate the probability of a particular class and make classifications based on a threshold value. For example, if the probability exceeds a cutoff (e.g., 0.5), the model assigns the input to that class. Because logistic regression requires a binary outcome variable (e.g., “TRUE” or “FALSE”), we will exclude the “star” category and limit our analysis to just quasars and galaxies. This change also makes it easier for us to compare our model with those from earlier research, a comparison that will be discussed later.

We will modify our training and testing to contain just “quasar” and “galaxy” observations. Using this refined dataset, we will apply the “glm” function, with “classlabel\_dsc\_joint” as the outcome variable and “phot\_rp\_mean\_mag” as the predictor. “Phot\_rp\_mean\_mag” is chosen as the predictor because it was the first decisive split in our previous decision tree model, making it a strong candidate for predicting the class in the logistic regression model.

*Glm Output:*

Coefficients:

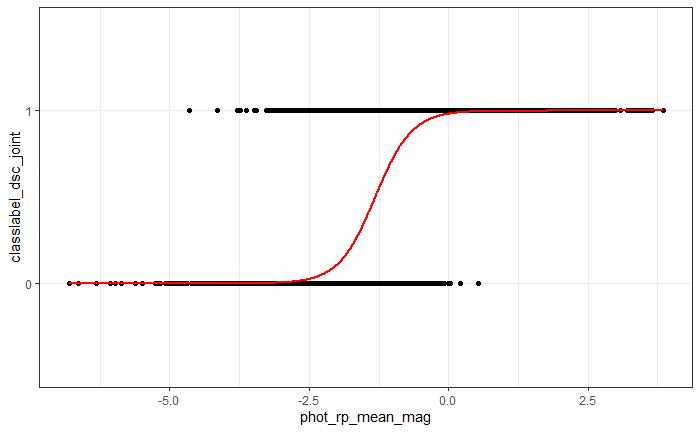
Estimate Std. Error z value Pr(>|z|)

(Intercept) 3.99730 0.04211 94.93 <2e-16 \*\*\*

phot\_rp\_mean\_mag 3.04666 0.03370 90.40 <2e-16 \*\*\*

The output shows that our predictor is statistically significant (<2e-16), with an estimate of ~3.04. This indicates that as “phot\_rp\_mean\_mag” increases, the likelihood of being classified as a quasar increases significantly. To visualize this probability increase, we will create a line plot with “phot\_rp\_mean\_mag” and “classlabel\_dsc\_joint” along with the predicted probabilities from the “glm” function.

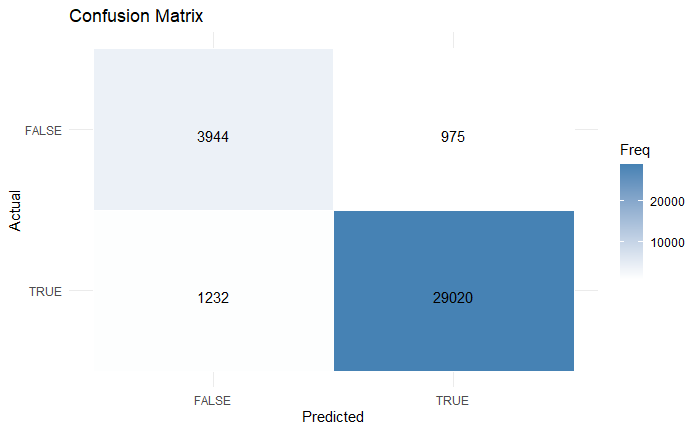
*Logistic Plot Below:*



Looking at the graph, we observe that the probability of the object being a quasar begins to increase as “phot\_rp\_mean\_mag” moves between -2.5 and 0. The slope of the line, as indicated by the “glm” function, is approximately 3.04. However, this value should be interpreted with caution. Since the function is a logistic regression, we must exponentiate the slope to understand the true change in probability. Doing so gives us a value of ~ 21.04, meaning that for each one-unit increase in “phot\_rp\_mean\_mag”, the odds of the object being a quasar increase by a factor of ~21.04.

Using this model, we will now make predictions on our testing dataset and evaluate its performance with a confusion matrix.

*SLR Confusion Matrix Below:*



*FALSE: Galaxy, TRUE: Quasar*

We see that the model performs well overall, especially in identifying quasars. However, it tends to misclassify some galaxies as quasars, likely due to the overlapping values in the predictor variable. The simple logistic regression model achieves a high overall accuracy of 93.72%. This is slightly lower than our bagging model (95.34%), slightly higher than our KNN model (93.49%), and about 2% better than our decision tree model (91.57%).

Next, we will build a multiple logistic regression model. Like the simple version, it estimates probabilities for classification, but this time we’ll include multiple predictor variables. By incorporating more information, this model may capture more complex patterns in the data and potentially improve accuracy. However, not all combinations of predictors are beneficial. To identify the most effective set, we’ll use the “bestglm” function from the “bestglm” library, which selects the optimal combination based on the **A**kaike **I**nformation **C**riterion (AIC).

***Note:*** *Since “bestglm” evaluates many combinations of variables to find the best fit. The process can be time-consuming, especially with large datasets or a high number of predictors. This dataset took ~6-10 hours to complete.*

*Bestglm Variable Combination:*

Photometric Variables:

* “Phot\_g\_mean\_mag” (Mean Magnitude of G Band)
* “Phot\_rp\_mean\_mag” (Mean Magnitude of RP Band)
* “Pseudocolour\_error” (Error in the Pseudocolour)
* “G\_rp” (G Band - RP Band: Color)

Distance and Location Variables:

* “Ra\_error” (Error in the Right Ascension)
* “Dec” (Declination)
* “Parallax\_error” (Error in the Parallax)
* “B” (Galactic Longitude)

After finding the best combination of variables, we can put these as predictors in our “glm” function.

*Glm Output:*

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) 9.542e+00 2.940e-01 32.454 < 2e-16 \*\*\*

phot\_g\_mean\_mag 2.249e+05 8.783e+04 2.560 0.01046 \*

phot\_rp\_mean\_mag -3.780e+05 1.476e+05 -2.560 0.01046 \*

ra\_error 8.387e-01 3.044e-01 2.755 0.00587 \*\*

dec -2.657e-01 1.154e-01 -2.303 0.02129 \*

parallax\_error -8.260e-01 2.793e-01 -2.957 0.00310 \*\*

pseudocolour\_error -4.776e-01 3.117e-01 -1.532 0.12545

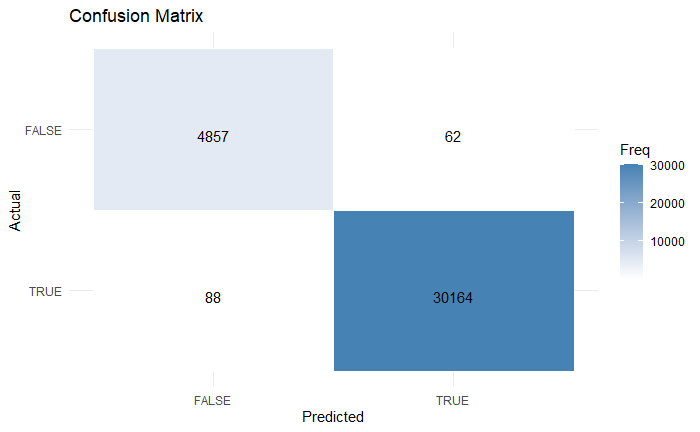
g\_rp -2.571e+05 1.004e+05 -2.560 0.01045 \*

b 1.386e-01 9.379e-02 1.478 0.13936

The output shows that several predictors are statistically significant (p < 0.05), including “phot\_g\_mean\_mag”, “phot\_rp\_mean\_mag”, “ra\_error”, “dec”, “parallax\_error”, and “g\_rp”. These variables contribute meaningfully to the classification decision. On the other hand, predictors like “pseudocolour\_error” and “b” are not significant, suggesting they may have less direct influence on the model’s performance, but their inclusion likely helps improve the AIC score.

Next, we will apply the model to make predictions and generate a confusion matrix.

*MLR Confusion Matrix Below:*



*FALSE: Galaxy, TRUE: Quasar*

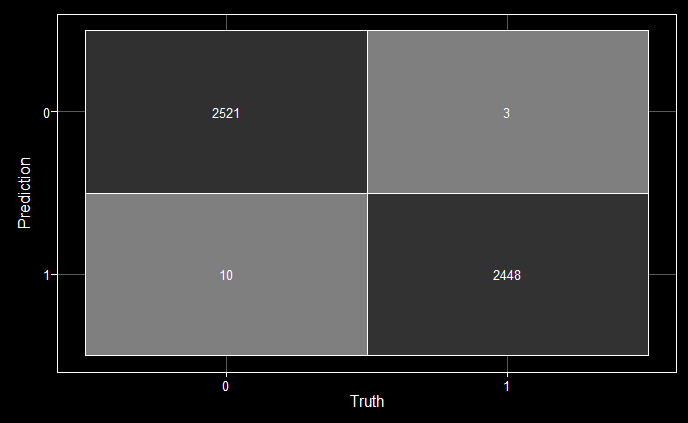
Overall, the model performs exceptionally well. The total accuracy of this model is 99.57%, which is significantly better than any of the other models (~4% Better than the previous best model). This raises the question: how does it compare to models developed in earlier research? In the following section, we’ll explore historical models to evaluate how far our approach has advanced.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

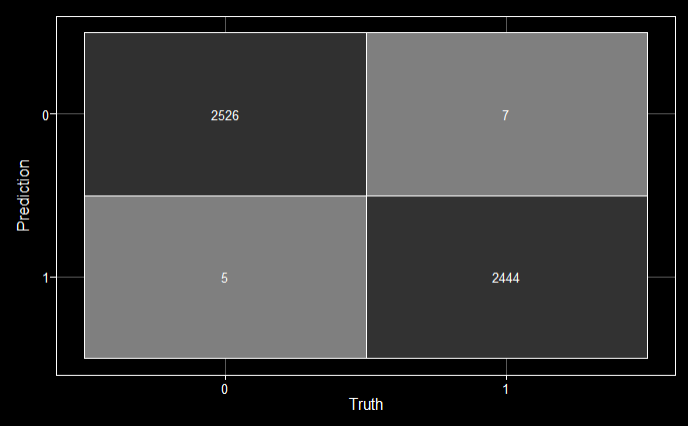
Compared to Historic Models

In this section, we compare the performance of our best model to that developed by Michael Kachanyuk’s group for the *Data in the Cosmos* Spring 2024 final project. Their team implemented three models: a multinomial logistic regression, a decision tree, and a random forest. Impressively, each achieved an accuracy above 99.70%, with the multinomial regression reaching 99.73%, the decision tree 99.75%, and the random forest reaching 99.81%.

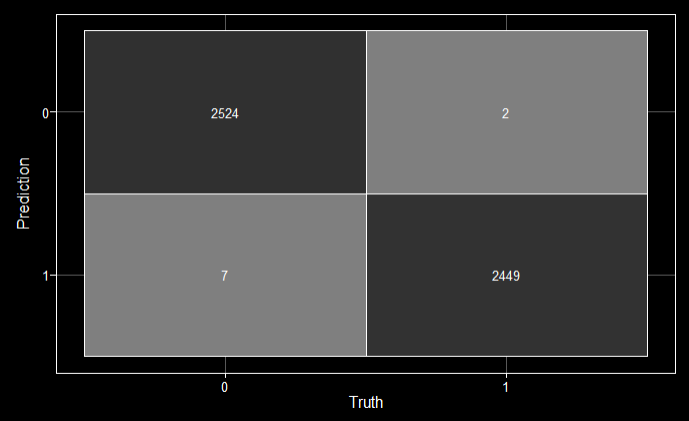
*Multiple Regression Confusion Matrix Below:*



*Decision Tree Confusion Matrix Below:*

**

*Random Forest Confusion Matrix Below:*

**

At first glance, one might argue that our model underperforms compared to the three historical models, especially given the longer runtime. However, a closer examination reveals a key distinction: the models developed were trained on a significantly smaller dataset (approximately 8,000 entries), whereas our model was trained on a much larger dataset of 80,000 entries. This suggests that our model is not only on par with them in terms of accuracy but may be more robust, achieving comparable performance while handling a tenfold increase in data volume.

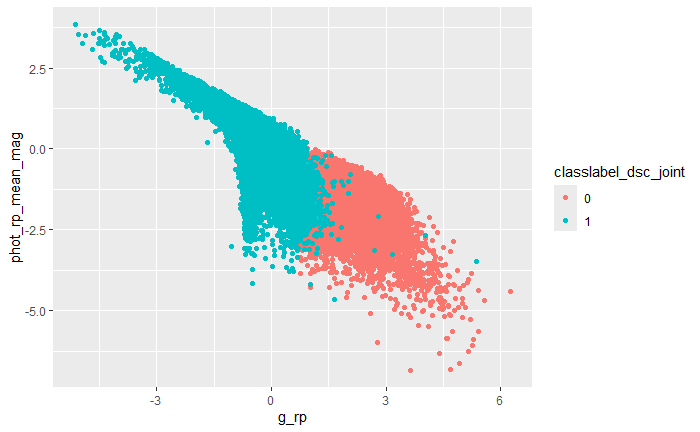
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Linear and Quadratic Discriminant

In our final section, we will explore **L**inear **D**iscriminant **A**nalysis (LDA) and **Q**uadratic **D**iscriminant **A**nalysis (QDA). The LDA model works by finding a linear projection that maximizes the distance between the class means while minimizing the variance within each class, effectively enhancing class separability in the projected space.

Initial testing with our training and testing set reveal collinearity among the predictor variables, which negatively impacts the performance of both LDA and QDA. To address this, we reduce our data to two variables: “phot\_rp\_mean\_mag” and “g\_rp”. These were selected based on their importance in both the decision tree model and the MLR model, where they appeared as key splits and had the lowest p-values. This makes them strong candidates for classification while avoiding issues with collinearity.

*Phot\_rp\_mean\_mag vs g\_rp Graph Below:*

**

*0: Galaxy, 1: Quasar*

To create a LDA model, we will use the function “lda” from the “MASS” library with our modified training set as the data.

*LDA Function Output Below:*

Call:

lda(classlabel\_dsc\_joint ~ ., data = space.train2)

Prior probabilities of groups:

0 1

0.139865 0.860135

Group means:

g\_rp phot\_rp\_mean\_mag

0 2.196558 -1.7771270

1 -0.353812 0.2874339

Coefficients of linear discriminants:

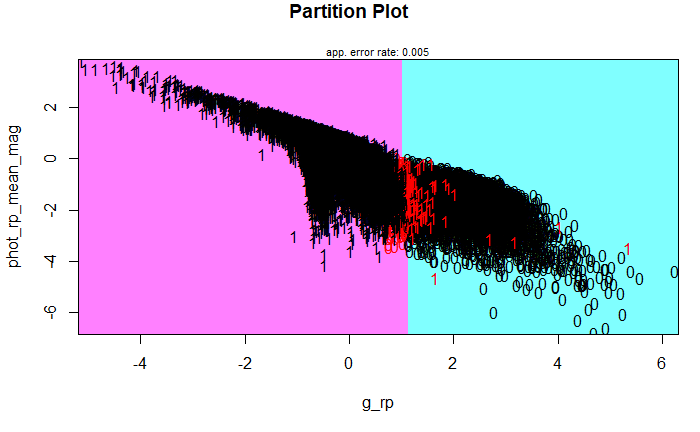
LD1

g\_rp -2.15139125

phot\_rp\_mean\_mag -0.01872338

Looking at the “group means” section of the output, we see that galaxies tend to have higher “g\_rp” values and lower “phot\_rp\_mean\_mag” values compared to quasars. With the model trained, we can now proceed to predict the classifications on the test set and visualize the decision boundaries to see how well LDA separates quasars from galaxies. We will use the library “klaR” which will help us visualize the boundaries.

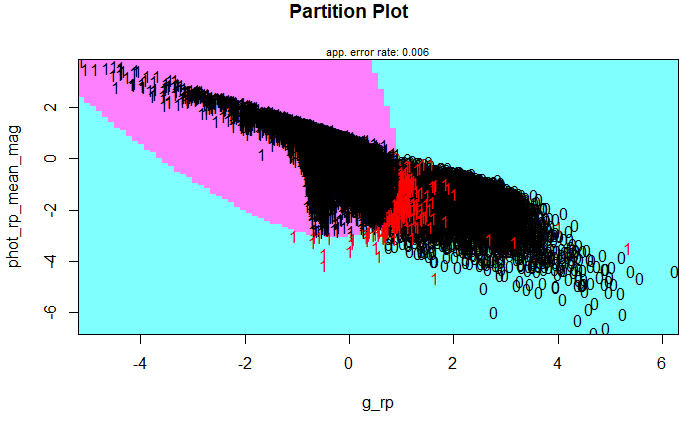
*LDA Decision Boundary Graph Below:*

**

Looking at the graph we can see that the LDA model divides the quasar and galaxy groups roughly at 1.25 on the “g\_rp” axis. This simple yet effective division results in a high accuracy of 99.51%. However, some quasar observations are classified as galaxies due to the curved overlap in the region. QDA could potentially address this by modeling the curved boundaries more effectively, leading to better classification accuracy.

Quadratic Discriminant Analysis works almost exactly the same way as LDA. Instead of fitting a *linear* boundary function, QDA uses a *quadratic* function to create the boundary. We will create a QDA model by using the function “qda” and our training set. This function has the same output as the “lda” function, so we will move onto predicting and visualizing the decision boundary.

*QDA Decision Boundary Graph Below:*



When we use QDA, we notice that more quasars are misclassified into the galaxy section due to the model fitting. This is reflected in the drop in accuracy, with the model achieving 99.3%, which is 0.21% lower than the LDA model’s accuracy. The quadratic function fits the data in such a way that it curves more sharply, capturing more quasars in the region around -0.25 on the “g\_rp” axis. This aggressive curve leads to more misclassifications among the quasars.

Both the LDA and QDA models perform well and achieve accuracies above 99%. The LDA model slightly outperformed the QDA with an accuracy of 99.51%, but both models fall just short of the multiple logistic regression model (99.57%). This suggests that the additional variables not included in the two models contribute a small amount of extra information, boosting the accuracy slightly. However, the difference between these models is minimal, and either model is a solid choice for classification.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Conclusion

All models examined achieve impressive accuracies exceeding 95%, with the multiple logistic regression model emerging as the top-performing model (99.57%). While it performed slightly below historic models in raw accuracy, it was trained on a nearly 10-20 times larger dataset. This suggests that, despite its longer runtime, the model is well-suited for large-scale applications and delivers competitive performance under more demanding conditions.