Stellar Classification

by Anthony Obrzut, Zibo Shang and Ai Yang

Section 1- Abstract

Background:
Stellar classification is the classification of stellar objects based on their spectral characteristics. As the technology behind telescopes had improved over time, the classification of stellar objects became increasingly more important to astronomers. Classifying galaxies, quasars, and stars is fundamental in modern astronomy because the distribution of stellar objects has helped us understand how the universe is made up. Our dataset was downloaded from Kaggle, with 100,000 observations of stellar objects taken by the Sloan Digital Sky Survey (SDSS). Each observation is described by 17 features and has 1 class feature that identifies a stellar object as either a galaxy, quasar, or star.

Methods:
Through an exploratory data analysis, some feature columns were transformed, and others were dropped due to their irrelevance. This step was followed by randomly selecting a subset of 10% of the processed data. We further split 80% of the subset as a training set and the other 20% as a testing set to test the performance of the proposed models initially. Multinomial Logistic Regression, K-Nearest Neighbors (KNN), and Random Forest models were constructed on this subset. The undropped feature columns were used to construct the Multinomial and Random Forest models. Two sets of eight variables were selected for the KNN method, one subset chosen based on a variable importance plot from our Random Forest Model and the other based on permutation feature importance. This resulted in two separate KNN models with different subsets of variables. The constructed models were then applied to the testing dataset, and their misclassification rate was compared based on their 50% and 80% prediction intervals. The superior KNN model was selected to be the random forest-based model according to its lower misclassification rate relative to the permutation-based KNN model.

Next, we took the whole dataset and did another 80% - 20% training - holdout split. In the 80% training set, we performed a three-fold cross-validation. We randomly chose one model from the threefold models. Then, we made predictions based on the 20% holdout set to obtain the performance of each model. The 50% and 80% prediction intervals were calculated based on each fold for different models. The misclassification rate was calculated and compared between each model and fold based on the prediction interval result.

Results:
Among the four constructed models for Phase B, the random forest model had the lowest average misclassification rate for both 50% and 80% prediction intervals on the testing dataset (50% PI:0.038, 80% PI: 0.018), followed by the multinomial model (50% PI: 0.053, 80%: 0.032), KNN-RF based model (50% PI: 0.154, 80%: 0.057) and KNN-permutation based model (50% PI: 0.238, 80%: 0.058).

In Phase C, the misclassification rates of the randomly chosen fold models were calculated, with the random forest model having the lowest misclassification rate, followed by the multinomial model and KNN-RF based model, respectively.
Similar results were obtained when using our models for prediction on the 20% holdout set. The misclassification rate for the Random Forest model is the lowest (50% PI:0.032, 80% PI: 0.015), followed by Multinomial model (50% PI:0.051, 80% PI: 0.029) and KNN-RF based model(50% PI:0.105, 80% PI: 0.038).

Conclusion:
After transforming our data and rigorously testing our proposed models, our group concluded that Random Forest is the most suitable model for our data due to its low misclassification rates, followed by Multinomial Logistic Regression and KNN Classification. The most critical variables consisted of redshift, different types of filters in the photometric system such as u, g, r, i, and z, and the spectrum identifiers such as plate and MJD.

Section 2 - EDA and Data Transformation

Variable Definitions

The initial dataset consists of 17 explanatory variables, which come from the observations taken from the SDSS. The variables are based on the images and the corresponding spectral characteristics of the stellar objects. The SDSS measures many spectra in a single observation, by means of a plate, an aluminum disk placed in the focal plane of the telescope. Each plate corresponds to a specific patch of sky, and is pre-drilled with holes corresponding to the sky positions of objects in that area, meaning that each area requires its own unique plate. A brief description of each variable listed below in table 1.

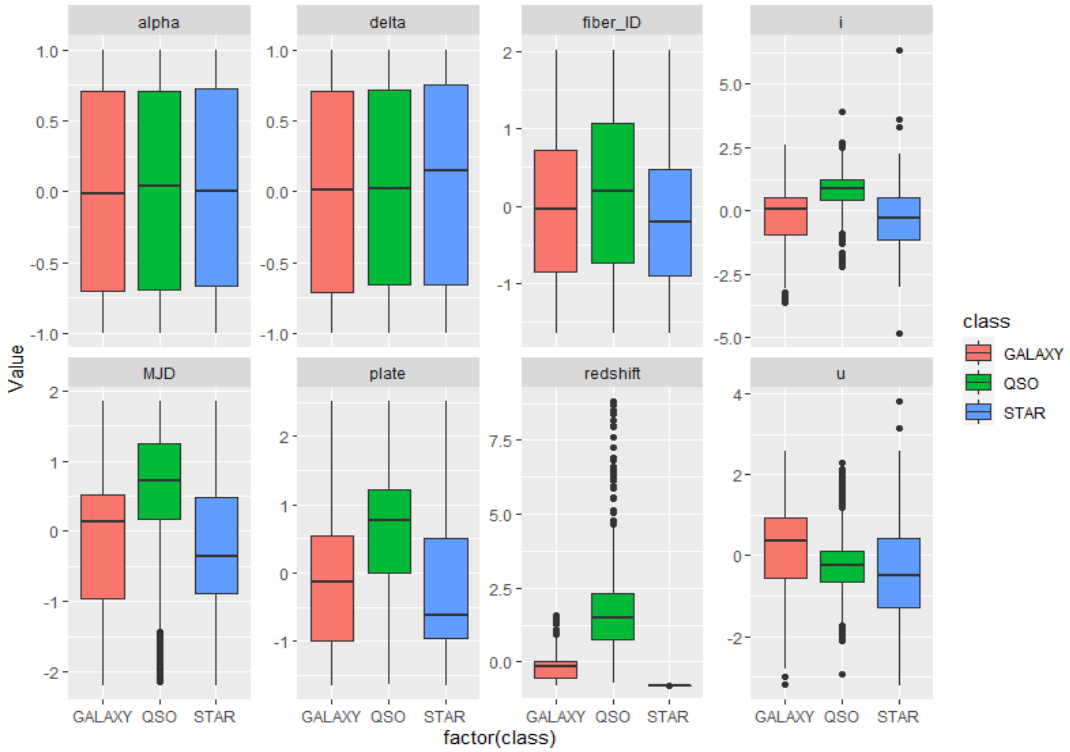
Our response variable, class, is categorized into three different groups: stars, which are fixed luminous objects in the sky, like the the sun in the Solar system; galaxies, which are systems of millions or even billions of stars held together by gravity; and quasars,

which are remote and extremely luminous stellar objects powered by a supermassive black hole. In our dataset, stars, galaxies and quasars are indicated as strings by STAR, GALAXY and QSO respectively.

Table 1. Definitions of variables

Spectral feature	alpha	Right ascension angle
	delta	Declination angle
	u	Ultraviolet filter in the photometric system
	g	Green filter in the photometric system
	r	Red filter in the photometric system
	i	Near infrared filter in the photometric system
	z	Infrared filter in the photometric system
	redshift	Redshift value based on the increase in wavelength based on the spectrum of the object
SDSS Image feature	cam_col	Camera column to identify the scanline within the run
	obj_ID	Object identifier, used to identify the object in the image
	run_ID	Run number used to identify the specific scan
	rerun_ID	rerun number to specify how the image was processed
	field_ID	Field number to identify each field
	spec_obj_ID	Unique id used to identify optical spectroscopic objects
SDSS Spectrum Identifiers	plate	Plate ID used to identify which SDSS plate is used to collect the spectrum
	MJD	Modified Julian Date, used to indicate when a given piece of SDSS data (the picture) was taken
	fiber_ID	fiber ID that identifies the fiber that pointed the light at the focal plane in each observation

Figure 1. Boxplot of the proposed important variables

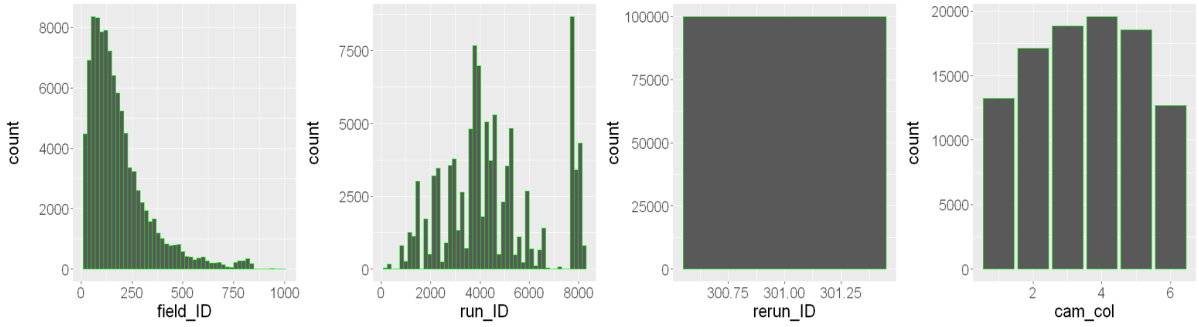


Based on the definition of each variable, we propose that the spectral features will be important when classifying the stellar objects, since galaxies, quasars and stars should exhibit distinguishable spectral characteristics between each other. Similarly, the spectrum identifiers are also expected to play key roles in classifying the stellar objects.

Data Transformation

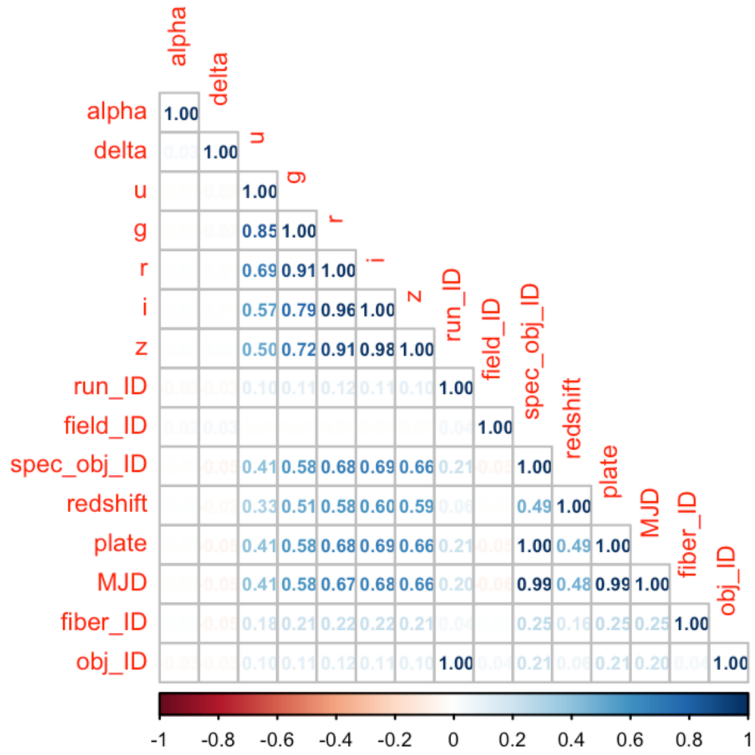
Before data transformation, we pruned one row of extreme values that has u, g and z equal to -9999, which gives us a dataset that consists of 99,999 observations with 17 feature columns and 1 response variable column.

Figure 2. Histograms of field_ID, run_ID, rerun_ID, cam_col



The histograms of each feature column have been plotted (shown in the appendix 1.1) in order to give us intuition behind the data transformation decisions we made. The angles, (variable name: alpha and delta) underwent a cosine transformation since the numbers are between 0 and 360 degrees, with 0 and 360 representing the same angle. The histograms of two variables, field_ID and run_ID both display highly skewed distributions. So, we binned them using suitable quantiles as the binning breaks. As a result, 5 bins were produced for field_ID and run_ID respectively to make the data relatively evenly distributed across all of the bins (shown in the appendix 1.2). This was followed by transforming the bins into categorical factors. rerun_ID was dropped since it has a constant value of 301 for every observation, and thus, would have no contribution to our classification models. cam_col and class were also transformed into categorical factors. Lastly, we performed feature scaling on each numerical variable to minimize the dominance of large distances in certain variables to make our distance based models viable.

Figure 3. Spearman correlation plot after initial data transformation



A spearman correlation plot (shown in Figure 3) was constructed after the first stage of data transformation to measure the strength and direction of monotonic association between each pair of numeric variables (shown in Figure 2). The correlation between many of the filter variables, namely u and g, g and r, r and i, i and z were close 1, suggesting that there exist strong linear relationships among the filter variables. So, we dropped columns g, r and z since they appeared to have the strongest spearman

correlation values with respect to other filter variables. In addition, plate was highly correlated with both MJD and spec_obj_ID. We constructed violin plots of these three variables to show their quantiles and data distribution density (shown in appendix 1.4). spec_obj_ID was dropped because its distribution is almost identical to the distribution of plate. The correlation between obj_ID and run_ID is 1. So, we dropped one of them to prevent any issues with collinearity arising in the future. obj_ID was chosen to be dropped because of its large scale of numerical values.

Section 3 - Initial Model Construction

Proposed Models

We decided to fit three models to our data; Multinomial Logistic Regression, K-Nearest Neighbors Classification (KNN) and a Random Forest model. The Multinomial Logistic Regression model was chosen due to its ease of interpretability regarding construction of prediction intervals, the model output indicating the important features from the data, and because it does not require many assumptions to be fulfilled by the data, such as normality. Random Forest was chosen because it often produces low misclassification rates, and is also suitable for prediction interval construction. The KNN model was chosen because of its ease of interpretation, which consists of assigning parameter values to the number of neighbors used for classification, and which distance function would be used, which is the Euclidean distance in our case. Our group was interested in observing the performances of each of these models because they differ greatly in their predicting methods, Multinomial Logistic Regression being a statistical model, Random Forest being a tree-based model, and KNN being a distance-based model.

In order to assess whether our three proposed models could be adequate for our final predictions, we took a subset of 10% of our dataset, which we then performed an 80% - 20% train - holdout split on. Both the Multinomial and Random Forest models were constructed with the variables that we had chosen in phase A, as shown below in Figure 4.

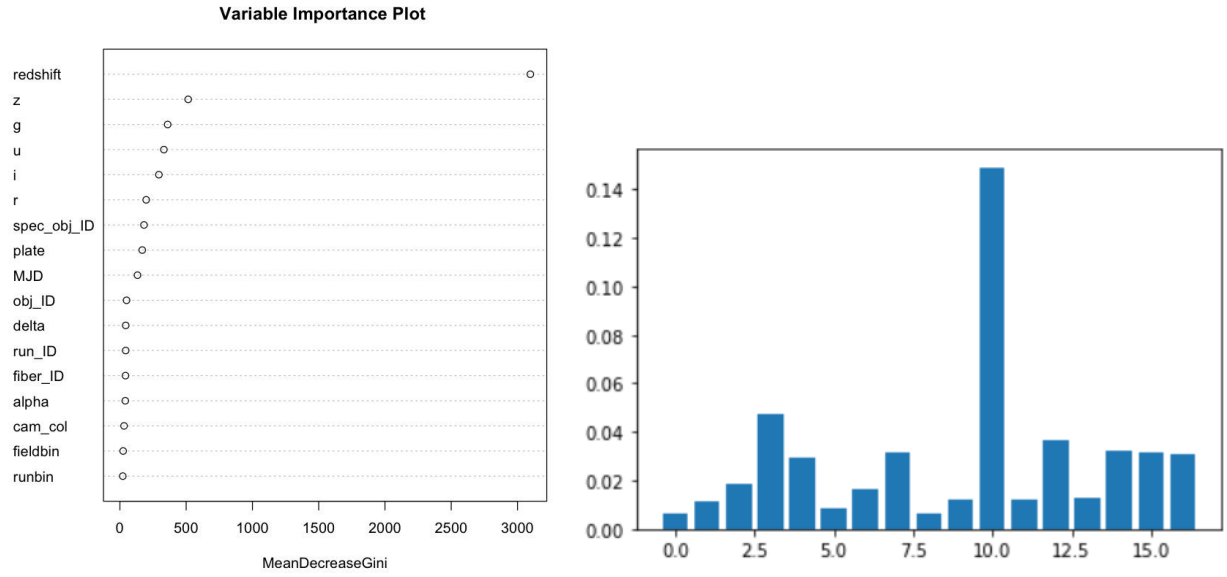
Figure 4. Selected Variables for Multinomial and RF models

```
Feature Subset For Multinomial and RF

'X'  'alpha'  'delta'  'u'  'i'  'redshift'  'plate'  'MJD'  'fiber_ID'  'cam_col'
'fieldbin'  'runbin'  'class'
```

Since KNN is not sensitive to collinearity between features, we decided to use two different methods for selecting sub-features in our KNN model. Our first method is based off of the variable importance plot of the Random Forest model on the full dataset as shown in Figure 5a. The top eight important variables are selected to construct the KNN-RF based model, which are redshift, z, g, u, i, r, spec_obj_ID, plate and MJD. The second method utilized permutation feature importance from the sklearn package in python, which first fits a model to the data, produces predictions on the data using that model, but the values of a feature in the dataset are scrambled. This is repeated a certain number of times until each feature is assigned a mean importance score. The larger the score, the greater the importance of the feature, which was measured by the prediction accuracy of the model. A plot of mean importance scores is shown in Figure 5b. For this KNN-permutation based subset selection, we selected the top eight important variables since we want to have the same number of variables as the RF-based KNN model. The selected variables are u, g, z, redshift, MJD, cam_col, fieldbin and runbin for our KNN-permutation based model construction.

Figure 5. a(left): variable importance plot of the random forest model based on a full dataset. **b(right):** mean importance plot for each feature.



Performance of Models:

Our models were constructed based on the chosen variable subsets. The results for each model were calculated based on the misclassification rates from 50% and 80% prediction intervals, as shown in appendix 2.1. In order to compare the performance of each model on the holdout dataset, we took the average of the misclassification rate on each class based on different prediction intervals. The results are displayed in figure 6.

Figure 6. Model comparison in phase B, based on the average misclassification rate.

	Multinomial	kNN-rf	kNN-perm	Random Forest
Avg 50% PI	0.053	0.154	0.238	0.038
Avg 80% PI	0.032	0.057	0.058	0.018

As shown in figure 6, the misclassification rate for the 80% prediction intervals in each of our models is smaller than the results of the 50% prediction intervals. This reduction in misclassification rate is much greater in our KNN models than the Multinomial and Random Forest model. Among our four models (including the two different KNN models with different subsets of variables), Random Forest has the smallest misclassification rate for both 50% and 80% prediction intervals. Among two KNN models, the one with the subset of variables selected from the Random Forest variable importance plot in figure 5a has a smaller misclassification rate. Thus we chose redshift, z, g, u, i, r, spec_obj_ID, plate and MJD as the subset to construct the KNN model in phase C.

Section 4 - Model Performance on Holdout Set

For each of our models, we performed a three fold cross-validation on our training set to estimate the level of fit of each of our models to our data. We then randomly chose a training-fold model for each of our prediction methods to apply to our holdout set, as the performance of each fold model should be similar since it only depends on the data contained in the folds. However, doing cross validation is still beneficial to our project, as it can reduce the runtime during model construction as well as remove the potential bias in the whole dataset. The misclassification rate table for each chosen training fold model is shown in figure 7.

Figure 7. a. Multinomial misclassification rates for the third training-fold model. b. Random Forest misclassification rates for the second training-fold model. c. KNN misclassification rates for the third training-fold model.

a. Multinomial model

	GALAXY	QSO	STAR
50% PI	0.033	0.128	0
80% PI	0.017	0.072	0

b. Random forest

	GALAXY	QSO	STAR
50% PI	0.019	0.078	0.001
80% PI	0.007	0.044	0.000

c. KNN model

	GALAXY	QSO	STAR
50% PI	0.044	0.106	0.173
80% PI	0.013	0.073	0.035

Model Comparison

After we had chosen our final models, we computed their misclassification rates on the holdout set. As expected, the misclassification rates we computed in phase B are quite similar to the misclassification rates we computed when we applied our models to both the cross-validation and the holdout set. Out of the three models we fitted to our data, we found that Random Forest had the lowest misclassification rates on average, deeming it the most accurate model. The multinomial model also performed relatively well. Even though the KNN model had slightly higher misclassification rates, we still consider KNN to be adequate for our data. An underlying reason why KNN may have not performed as well as the other two models is that KNN generally does not perform well on larger datasets such as ours.

Figure 8. a. Multinomial misclassification rates in the holdout set. b. Random Forest misclassification rates in the holdout set. c. KNN misclassification rates in the holdout set.

a. Multinomial model

	GALAXY	QSO	STAR
50% PI	0.031	0.122	0
80% PI	0.015	0.072	0

b. Random forest

	GALAXY	QSO	STAR
50% PI	0.016	0.079	0
80% PI	0.005	0.041	0

c. KNN model

	GALAXY	QSO	STAR
50% PI	0.040	0.099	0.177
80% PI	0.012	0.066	0.035

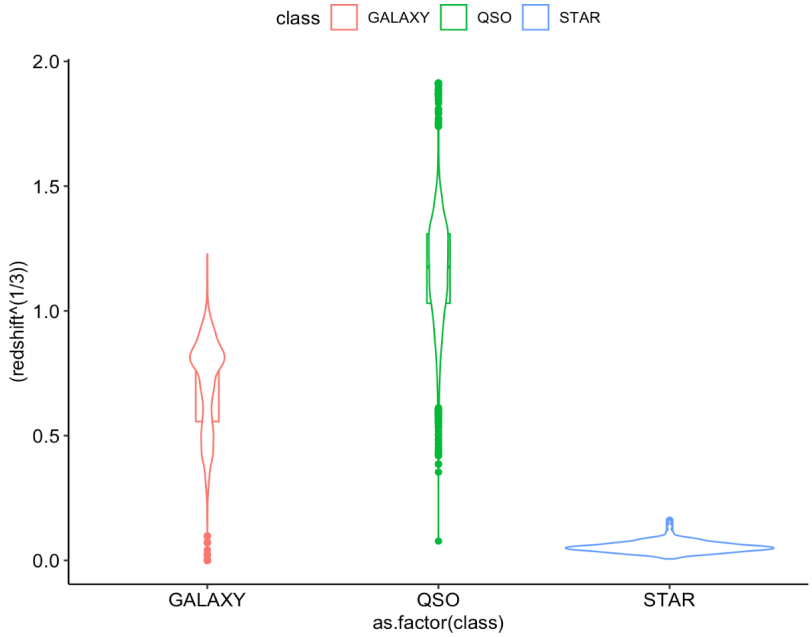
Figure 9. The misclassification rate comparison table for all models in phase C.

	Multinomial	kNN-rf	Random Forest
Avg 50% PI	0.051	0.105	0.032
Avg 80% PI	0.029	0.038	0.015

In each model, the STAR class had the lowest misclassification rates, suggesting that STAR may be the most distinguishable class in our response variable. It is important to note that Random Forest did not have any misclassifications for the STAR class. In addition, the multinomial model had a near-zero misclassification rate for the STAR class. On the other hand, the QSO class had the highest misclassification rates for all three models, suggesting that QSO may be the most indistinguishable class in our response variable. Referring back to the three boxplots of the redshift variable for each class in Figure 1, and the boxplot/violin plot in Figure 10, the STAR class appears to have a much lower median and a much smaller spread than the other classes, which may be a significant reason why STAR was easier for our models to classify than the other classes. QSO has the largest spread, which may be a contributing factor as to

why QSO was the most difficult for our models to classify. In addition, the filter variables, which consist of u, g, r, i and z also proved to play a significant role in classifying the stellar objects, as evidenced by the variable importance plot in Figure 5a, and both MJD and plate proved to be important as well.

Figure 10. New boxplot/violin plot of the cubic root of the redshift variable based on different classes.



When comparing the results from the cross-validation and holdout set results for each model, the misclassification rates on GALAXY and QSO displayed small improvements, and the misclassification rates for STAR stayed relatively unchanged since they were already either 0 or close to 0 depending on the model. This could be indicative that the cross-validation models were neither underfitting or overfitting the data.

For all of our final models, both the cross-validation and holdout set results are quite similar to the results we obtained in phase B, which suggests that our phase B models were a good indication of what results we would obtain results from the cross-validation and holdout set.

Table 2. R Libraries used for each method

Model	**R library**
Multinomial	library(VGAM)
KNN	library(caret)
	library(class)
Random Forest	library(randomForest)

Conclusion

Our group wanted to test the performances of a statistical model, a tree-based model and a distance-based model. So, we selected Multinomial Logistic Regression, Random Forest and KNN Classification as our proposed models. After dropping the unnecessary and irrelevant columns from our data set, we proceeded to transform our data so that it would be suitable for fitting models. We first took a subset of 10% of our transformed data, which we then performed an 80% - 20% train - holdout split on and tested the performance of each model on this subset to give us an idea of how we should expect our final models to perform. Then, we split our original transformed data into training and holdout sets, which consisted of 80% and 20%, respectively, of the original transformed data. To estimate the fit of our proposed models on the data, we performed a three-fold cross validation on our training set from and randomly chose a training-fold for each model to test on our holdout set for our final misclassification rates.

Out of the three different methods our group used to predict the class of the stellar objects in our dataset, Random Forest had the best performance with the lowest misclassification rates, followed by Multinomial Logistic Regression and KNN Classification respectively. The results align with our group's expectations because Random Forest often performs well on many different datasets, Multinomial models are known to be reliable in multi-class classification, and KNN usually suffers higher misclassification relative to other models when fitted to large datasets, like ours.

Redshift appeared to be the most significant feature. This may be due to the fact that for each of our three classes; GALAXY, QSO and STAR, the respective distributions of redshift values were quite different, making it easier for our models to distinguish what class each observation should be assigned to on unseen data. Furthermore, our filter variables; u, g, r, i and z also appeared to be important for the prediction of the stellar objects.

Section 5 - Contribution:

Zibo Shang: Code integration and edition; Report proofreading

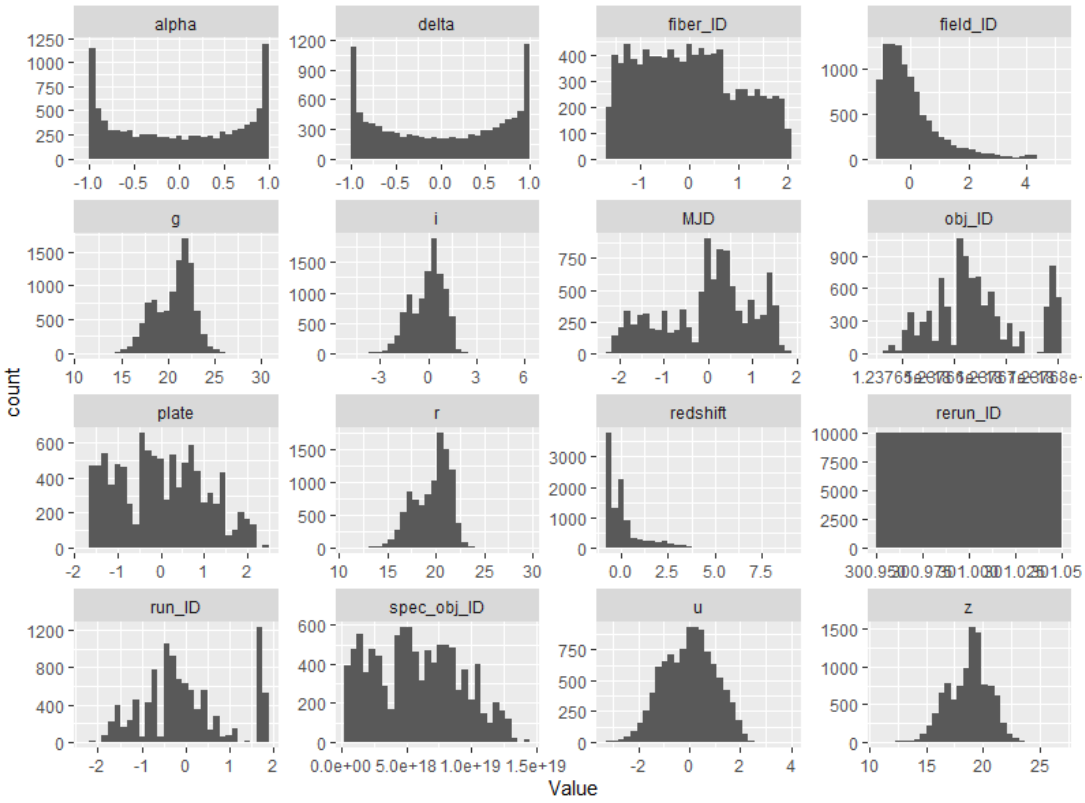
Anthony Obrzut: Report write-up and proofreading, code advisory for Zibo

Ai Yang: Report write-up and proofreading, code advisory for Zibo

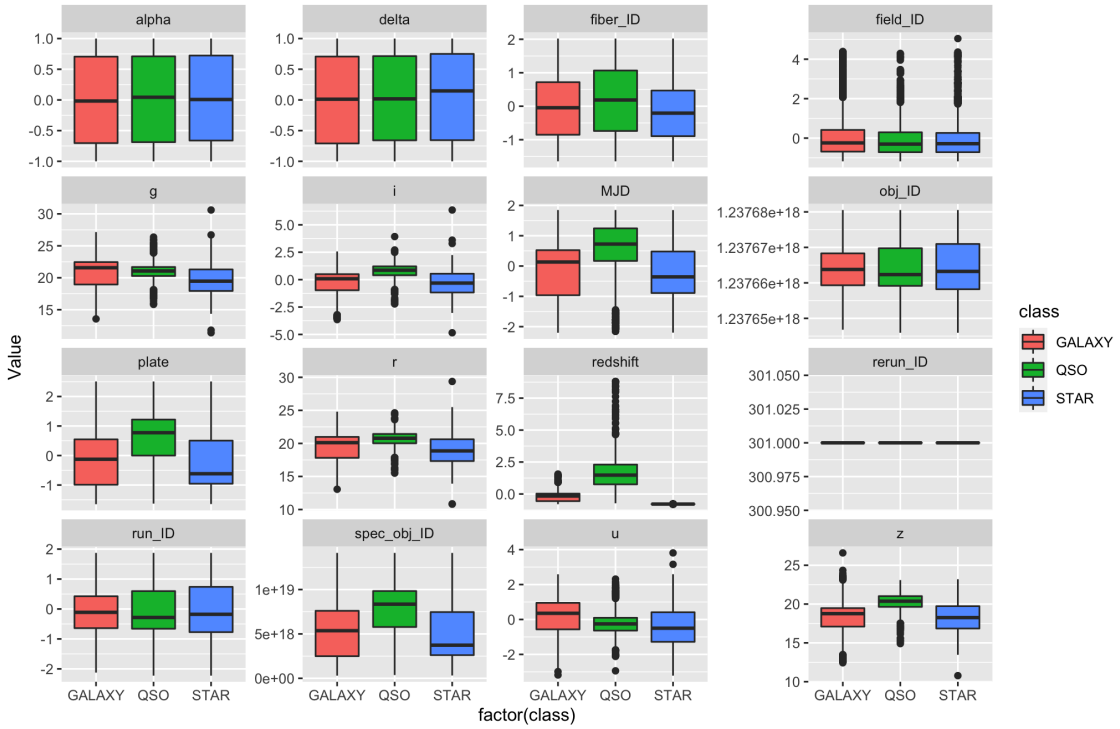
Appendix:

1. Figures and plots

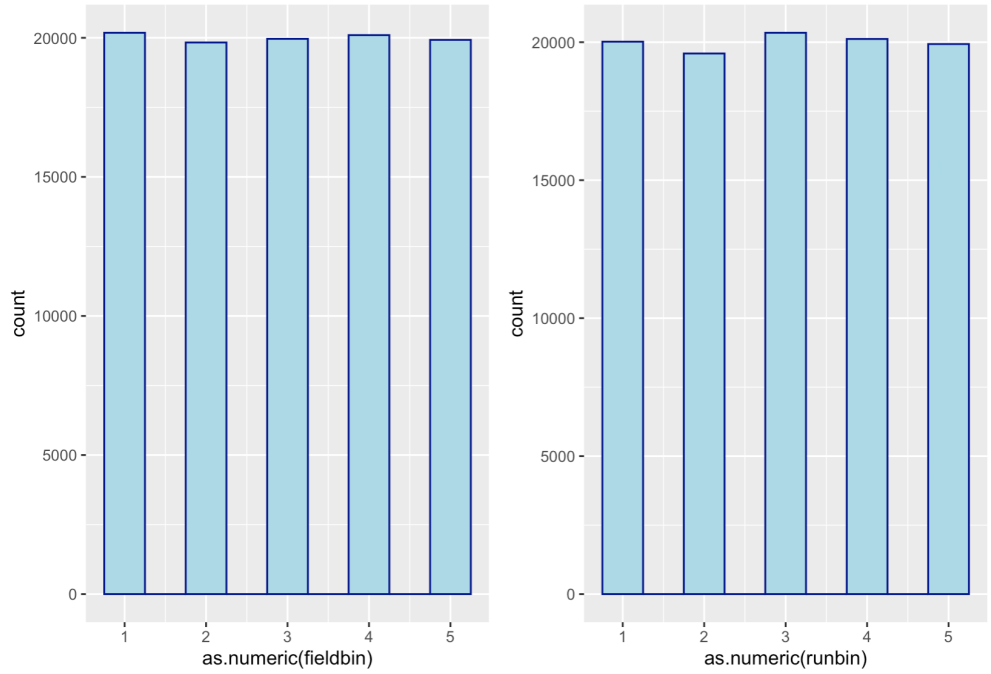
1.1 Histograms for all feature columns



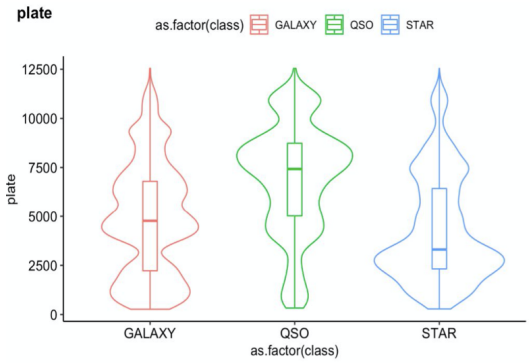
1.2 boxplot for all feature columns

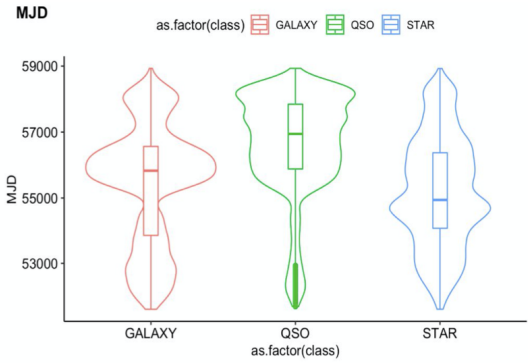


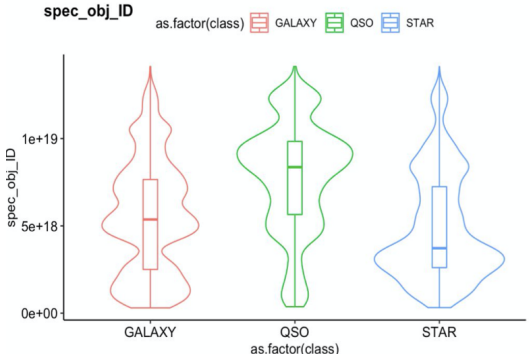
1.3 Distribution of field_ID and run_ID after binning



1.4 Violin plot of plate, MJD, spec_obj_ID







1.5 permutation importance for feature selection

```
Feature: 0, Score: 0.00634
Feature: 1, Score: 0.01108
Feature: 2, Score: 0.01846
Feature: 3, Score: 0.04732
Feature: 4, Score: 0.02906
Feature: 5, Score: 0.00878
Feature: 6, Score: 0.01612
Feature: 7, Score: 0.03166
Feature: 8, Score: 0.00634
Feature: 9, Score: 0.01218
Feature: 10, Score: 0.14911
Feature: 11, Score: 0.01218
Feature: 12, Score: 0.03642
Feature: 13, Score: 0.01252
Feature: 14, Score: 0.03192
Feature: 15, Score: 0.03122
Feature: 16, Score: 0.03052
```

2. Tables

2.1 Table 2. The misclassification rate for each model, calculated with 50% and 80% prediction intervals during phase B.

A.multinomial model

	GALAXY	QSO	STAR
50% PI	0.036	0.123	0
80% PI	0.019	0.075	0

B. random forest model

	GALAXY	QSO	STAR
50% PI	0.025	0.088	0
80% PI	0.013	0.040	0

C.RF-based KNN model

	GALAXY	QSO	STAR
50% PI	0.045	0.128	0.289
80% PI	0.012	0.076	0.083

D.permutation-based KNN model

	GALAXY	QSO	STAR
50% PI	0.018	0.159	0.537
80% PI	0.001	0.087	0.087