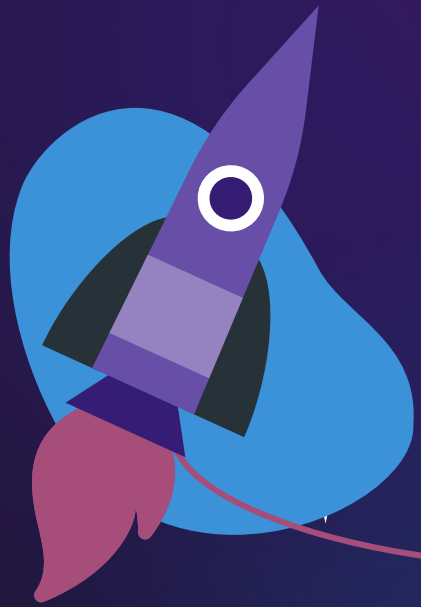
Stellar Classification - Phase 2

Anthony Obrzut
Ai Yang
Zibo Shang



Project Review

Project description: Classification of stellar objects based on their spectral characteristics.

Project importance: The classification scheme of galaxies, quasars, and stars is fundamental in astronomy. Understanding their distribution helps us understand how our universe is made up.

Dataset description: The data consists of 100,000 observations of space taken by the SDSS(Sloan Digital Sky Survey). Every observation is described by 17 feature columns.

Response variable: Class - either a star, galaxy or quasar.

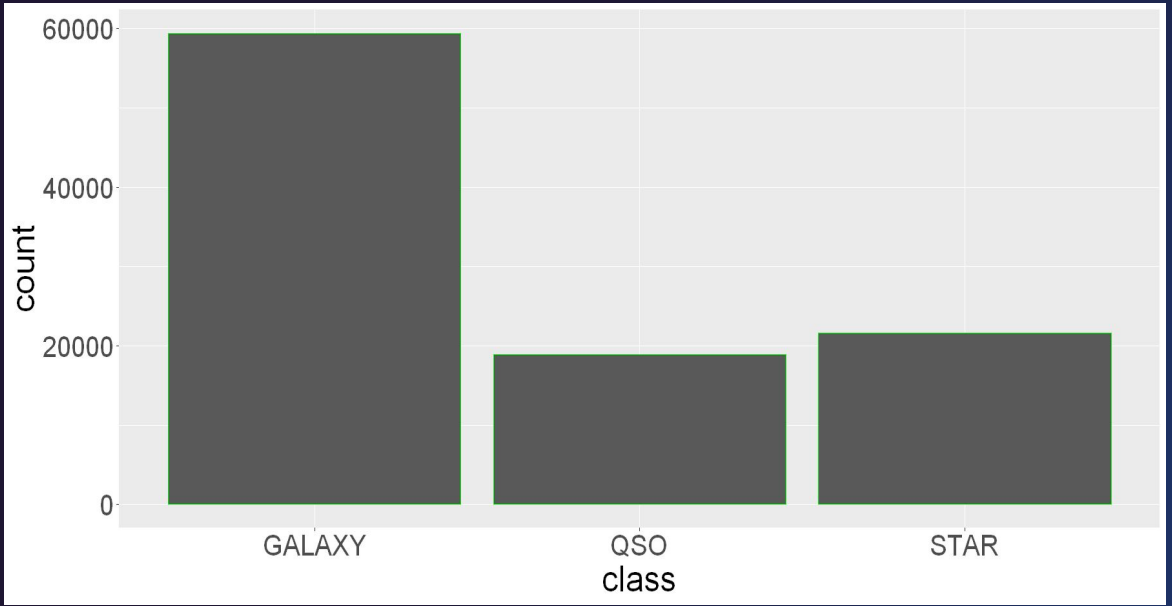
Goal: Correctly classify stars, galaxies, and quasars based on their spectral characteristics.

Review: Transformation and Wrangling

Table I. Transformation Decision		
Variable Type	Name	Decision
Categorical	Class*	Covert Strings into numeric response
Numeric(dbl)	alpha	Cosine Transformation
Numeric(dbl)	delta	Cosine Transformation
Numeric(dbl)	i	Standardization
Numeric(dbl)	u	Standardization
Numeric(dbl)	g	DROP, high correlation
Numeric(dbl)	r	DROP, high correlation
Numeric(dbl)	z	DROP, high correlation
Numeric(int)	obj_ID	DROP, high correlation
Numeric(int)	cam_col	trans to categorical by factor
Numeric(int)	rerun_ID	DROP, same for all
Numeric(int)	spec_obj _ID	DROP, meaningless by definition
Numeric(int)	field_ID	Standardization and Binning
Numeric(dbl)	redshift	Standardization
Numeric(int)	run_ID	Standardization and Binning
Numeric(int)	plate	Standardization
Numeric(int)	MJD	Standardization
Numeric(int)	fiber_ID	Standardization

Undersampling Based on QSO

- The GALAXY class outnumbers the QSO and STAR classes by a large amount.
- Apply undersampling on the dataset based on the relative frequency of observations in the QSO class.
- Our original dataset has 99,999 observations, and our undersampled dataset has 56,883 observations.



Proposed Models (3 fold CV)

Three types of response variable

Unbalance response variable

Multinomial logistic regression

KNN

Random forest

Multinomial Model

- First, I constructed 3 full multinomial models, This was to create a baseline to compare my subsetted models. Based on the output of each full model, I decided to use delta, u, i, cam_col, redshift, plate, MJD, fieldbin, and runbin as my subset. I am contemplating taking delta out, as it does not appear to be as significant as the other variables

MODEL 1

```
Call:
vglm(formula = class ~ ., family = multinomial(), data = rbind(trainDat2,
    trainDat3))

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept):1  54.268425   1.644063  33.009  < 2e-16 ***
(Intercept):2  52.501495   1.658539  31.655  < 2e-16 ***
alpha:1        -0.014787   0.079012  -0.187  0.85154
alpha:2        -0.035802   0.087953  -0.407  0.68396
delta:1        -0.134471   0.081501  -1.650  0.09896 .
delta:2        -0.100244   0.090666  -1.106  0.26888
u:1            -0.644968   0.103973  -6.203 5.53e-10 ***
u:2            -3.224984   0.114346 -28.204  < 2e-16 ***
i:1             0.115477   0.088957   1.298  0.19425
i:2             1.662019   0.101999  16.295  < 2e-16 ***
run_ID:1        0.186309   0.243730   0.764  0.44462
run_ID:2       -0.036738   0.279594  -0.131  0.89546
cam_col2:1     -0.350658   0.200075  -1.753  0.07967 .
cam_col2:2     -0.223063   0.224664  -0.993  0.32077
cam_col3:1     -0.056261   0.191197  -0.294  0.76856
cam_col3:2      0.136395   0.215074   0.634  0.52597
cam_col4:1     -0.144943   0.192693  -0.752  0.45193
cam_col4:2      0.001822   0.216365   0.008  0.99328
cam_col5:1     -0.251482   0.203139  -1.238  0.21572
cam_col5:2     -0.271034   0.227006  -1.194  0.23250
cam_col6:1     -0.531222   0.219201  -2.423  0.01537 *
cam_col6:2     -0.343389   0.245566  -1.398  0.16201
field_ID:1     -0.251334   0.115869  -2.169  0.03007 *
field_ID:2     -0.188472   0.135599  -1.390  0.16455
redshift:1     74.687824   2.044840  36.525  < 2e-16 ***
redshift:2     78.336788   2.046986  38.269  < 2e-16 ***
plate:1         2.219711   0.265492   8.361  < 2e-16 ***
plate:2         1.605436   0.289316   5.549 2.87e-08 ***
MJD:1          -2.392164   0.255576  -9.360  < 2e-16 ***
MJD:2          -2.077445   0.282405  -7.356 1.89e-13 ***
fiber_ID:1      0.118636   0.072967   1.626  0.10397
fiber_ID:2      0.082304   0.078365   1.050  0.29360
fieldbin(-0.774,-0.45]:1  0.164030   0.185257   0.885  0.37593
fieldbin(-0.774,-0.45]:2  0.204172   0.205661   0.993  0.32083
fieldbin(-0.45,-0.06]:1   0.059267   0.201430   0.294  0.76858
fieldbin(-0.45,-0.06]:2  -0.086712   0.225181  -0.385  0.70018
fieldbin(-0.06,0.58]:1    0.382698   0.224173   1.707  0.08779 .
fieldbin(-0.06,0.58]:2    0.105061   0.255079   0.412  0.68043
fieldbin(0.58,5.4]:1      0.947590   0.344552   2.750  0.00596 **
fieldbin(0.58,5.4]:2      0.632379   0.395550   1.599  0.10988
runbin(-0.831,-0.327]:1   0.146761   0.259251   0.566  0.57133
runbin(-0.831,-0.327]:2   0.415629   0.290514   1.431  0.15253
runbin(-0.327,0.048]:1    0.532569   0.326460   1.631  0.10282
runbin(-0.327,0.048]:2    0.689589   0.370401   1.862  0.06264 .
runbin(0.048,0.8]:1       0.470356   0.450734   1.044  0.29670
runbin(0.048,0.8]:2       0.282023   0.510891   0.552  0.58093
runbin(0.8,1.9]:1        -0.145444   0.707919  -0.205  0.83722
runbin(0.8,1.9]:2         0.428875   0.817125   0.525  0.59968
```

MODEL 2

```
Call:
vglm(formula = class ~ ., family = multinomial(), data = rbind(trainD
    trainDat3))

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept):1  54.51287    1.69350  32.189  < 2e-16 ***
(Intercept):2  52.57577    1.70750  30.791  < 2e-16 ***
alpha:1        -0.07546    0.07940  -0.950 0.341926
alpha:2        -0.05776    0.08846  -0.653 0.513817
delta:1        -0.17568    0.08125  -2.162 0.030595 *
delta:2        -0.15043    0.09049  -1.662 0.096414 .
u:1            -0.57076    0.10128  -5.636 1.74e-08 ***
u:2            -3.18887    0.11209 -28.449  < 2e-16 ***
i:1             0.05742    0.08804   0.652 0.514318
i:2             1.53746    0.10071  15.267  < 2e-16 ***
run_ID:1        0.35652    0.24393   1.462 0.143861
run_ID:2        0.00651    0.27806   0.023 0.981321
cam_col2:1     -0.30623    0.20100  -1.524 0.127623
cam_col2:2     -0.13467    0.22629  -0.595 0.551753
cam_col3:1     -0.13440    0.19495  -0.689 0.490563
cam_col3:2     -0.02162    0.21997  -0.098 0.921686
cam_col4:1     -0.28494    0.19893  -1.432 0.152046
cam_col4:2     -0.15165    0.22296  -0.680 0.496393
cam_col5:1     -0.23906    0.20348  -1.175 0.240053
cam_col5:2     -0.20529    0.22854  -0.898 0.369046
cam_col6:1     -0.50378    0.22103  -2.279 0.022652 *
cam_col6:2     -0.35334    0.24835  -1.423 0.154811
field_ID:1     -0.27260    0.11526  -2.365 0.018025 *
field_ID:2     -0.29834    0.13627  -2.189 0.028567 *
redshift:1     74.59415    2.09818  35.552  < 2e-16 ***
redshift:2     78.28730    2.10033  37.274  < 2e-16 ***
plate:1         2.43158    0.26487   9.180  < 2e-16 ***
plate:2         1.79501    0.28866   6.218 5.02e-10 ***
MJD:1          -2.53818    0.25957  -9.778  < 2e-16 ***
MJD:2          -2.16507    0.28637  -7.560 4.02e-14 ***
fiber_ID:1      0.05362    0.07211   0.744 0.457132
fiber_ID:2      0.09677    0.07758   1.247 0.212313
fieldbin(-0.774,-0.45]:1  0.17167    0.18648   0.921 0.357276
fieldbin(-0.774,-0.45]:2  0.17432    0.20694   0.842 0.399586
fieldbin(-0.45,-0.06]:1   0.24248    0.20077   1.208 0.227136
fieldbin(-0.45,-0.06]:2  -0.01143    0.22558  -0.051 0.959599
fieldbin(-0.06,0.58]:1    0.43566    0.22570   1.930 0.053571 .
fieldbin(-0.06,0.58]:2    0.36150    0.25658   1.409 0.158865
fieldbin(0.58,5.4]:1      1.14101    0.34036   3.352 0.000801 ***
fieldbin(0.58,5.4]:2      0.98562    0.39425   2.500 0.012421 *
runbin(-0.831,-0.327]:1  -0.06822    0.25684  -0.266 0.790539
runbin(-0.831,-0.327]:2   0.29588    0.28823   1.027 0.304647
runbin(-0.327,0.048]:1    0.25496    0.32171   0.793 0.428057
runbin(-0.327,0.048]:2    0.48937    0.36525   1.340 0.180305
runbin(0.048,0.8]:1       0.05510    0.44640   0.123 0.901765
runbin(0.048,0.8]:2       0.14758    0.50593   0.292 0.770519
runbin(0.8,1.9]:1        -0.77863    0.71033  -1.096 0.273016
runbin(0.8,1.9]:2         0.10681    0.81459   0.131 0.895679
```

```
Call:
vglm(formula = class ~ ., family = multinomial(), data = rbind(trainDat1,
    trainDat2))

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept):1  54.05209    1.63674  33.024  < 2e-16 ***
(Intercept):2  52.55883    1.65086  31.837  < 2e-16 ***
alpha:1         0.03111    0.07839   0.397  0.69153
alpha:2         0.03009    0.08757   0.344  0.73109
delta:1        -0.12900    0.08103  -1.592  0.11138
delta:2        -0.11067    0.09038  -1.224  0.22078
u:1            -0.50221    0.09608  -5.227 1.72e-07 ***
u:2            -3.19166    0.10708 -29.805  < 2e-16 ***
i:1             0.04372    0.08491   0.515  0.60665
i:2             1.45616    0.09632  15.118  < 2e-16 ***
run_ID:1       -0.02025    0.24669  -0.082  0.93457
run_ID:2       -0.10359    0.28019  -0.370  0.71159
cam_col2:1     -0.42054    0.20090  -2.093  0.03632 *
cam_col2:2     -0.26434    0.22651  -1.167  0.24319
cam_col3:1     -0.08180    0.19034  -0.430  0.66737
cam_col3:2     -0.03428    0.21558  -0.159  0.87365
cam_col4:1     -0.18422    0.19216  -0.959  0.33772
cam_col4:2      0.03670    0.21613   0.170  0.86515
cam_col5:1     -0.48000    0.20627  -2.327  0.01996 *
cam_col5:2     -0.61556    0.23077  -2.667  0.00764 **
cam_col6:1     -0.68292    0.22140  -3.085  0.00204 **
cam_col6:2     -0.53911    0.24790  -2.175  0.02965 *
field_ID:1     -0.20130    0.11298  -1.782  0.07481 .
field_ID:2     -0.12637    0.13410  -0.942  0.34603
redshift:1     74.57288    2.04184  36.522  < 2e-16 ***
redshift:2     78.33318    2.04401  38.323  < 2e-16 ***
plate:1         2.20922    0.25643   8.615  < 2e-16 ***
plate:2         1.38864    0.28075   4.946 7.57e-07 ***
MJD:1          -2.34042    0.24830  -9.426  < 2e-16 ***
MJD:2          -1.75091    0.27566  -6.352 2.13e-10 ***
fiber_ID:1      0.09714    0.07196   1.350  0.17709
fiber_ID:2      0.15108    0.07742   1.951  0.05102 .
fieldbin(-0.774,-0.45]:1  0.25491    0.18635   1.368  0.17134
fieldbin(-0.774,-0.45]:2  0.29390    0.20671   1.422  0.15509
fieldbin(-0.45,-0.06]:1   0.02453    0.20483   0.120  0.90469
fieldbin(-0.45,-0.06]:2  -0.22274    0.22879  -0.974  0.33028
fieldbin(-0.06,0.58]:1    0.40228    0.22223   1.810  0.07026 .
fieldbin(-0.06,0.58]:2    0.17232    0.25397   0.679  0.49745
fieldbin(0.58,5.4]:1      0.98173    0.33915   2.895  0.00380 **
fieldbin(0.58,5.4]:2      0.63570    0.39352   1.615  0.10622
runbin(-0.831,-0.327]:1   0.48178    0.25745   1.871  0.06130 .
runbin(-0.831,-0.327]:2   0.61109    0.28809   2.121  0.03391 *
runbin(-0.327,0.048]:1    0.76340    0.32812   2.327  0.01999 *
runbin(-0.327,0.048]:2    0.66050    0.37003   1.785  0.07427 .
runbin(0.048,0.8]:1       0.78416    0.45404   1.727  0.08416 .
runbin(0.048,0.8]:2       0.31074    0.51151   0.608  0.54352
runbin(0.8,1.9]:1         0.41049    0.71665   0.573  0.56679
runbin(0.8,1.9]:2         0.43212    0.81802   0.528  0.59732
```

Multinomial Model, Misclassification Rates

- After constructing my subsetted models, I wanted to compare the misclassification performances between subsetted multinomial models on the undersampled dataset and the original dataset.

Multinomial No Undersampling Missclassification Rates Fold 1

	GALAXY	QSO	STAR
50%	0.016	0.124	0
80%	0.015	0.071	0

Multinomial No Undersampling Missclassification Rates Fold 2

	GALAXY	QSO	STAR
50%	0.023	0.131	0.001
80%	0.029	0.074	0.000

Multinomial No Undersampling Missclassification Rates Fold 3

	GALAXY	QSO	STAR
50%	0.030	0.133	0
80%	0.013	0.075	0

Multinomial Undersampled Missclassification Rates Fold 1

	GALAXY	QSO	STAR
50%	0.072	0.081	0
80%	0.027	0.040	0

Multinomial Undersampled Missclassification Rates Fold 2

	GALAXY	QSO	STAR
50%	0.071	0.081	0
80%	0.026	0.038	0

Multinomial Undersampled Missclassification Rates Fold 3

	GALAXY	QSO	STAR
50%	0.066	0.086	0
80%	0.025	0.044	0

Average Missclassification Rates Original Data

Galaxy	QSO	STAR
0.021	0.101	0

Average Missclassification Rates Undersampling

Galaxy	QSO	STAR
0.048	0.062	0

Multinomial Model, Best Model

- I chose the model on the 2nd fold to be the “best” model.
- We can rule out the 1st model because all respective misclassification rates are higher in the 1st model than the 2nd model, excluding STAR.
- The 2nd model slightly edges out thes 3rd model based on misclassification rates. They are quite similar, though.

Multinomial Undersampled Missclassification Rates Fold 1

	GALAXY	QSO	STAR
50%	0.072	0.081	0
80%	0.027	0.040	0

Multinomial Undersampled Missclassification Rates Fold 2

	GALAXY	QSO	STAR
50%	0.071	0.081	0
80%	0.026	0.038	0

Multinomial Undersampled Missclassification Rates Fold 3

	GALAXY	QSO	STAR
50%	0.066	0.086	0
80%	0.025	0.044	0

KNN Model

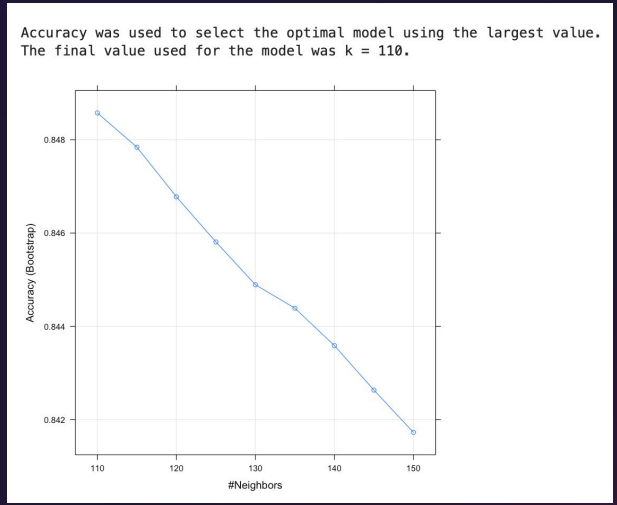
- Package: caret (Classification And REgression Training), similar to scikit-learn
- Number of neighbour: hyper-parameter, 132, Toning performed by seq(110,150,5)
- Distance Method: Standardization & factor levels (1,2,3,4...)

Variable Subset, only spectral values and significant non-significant variables are selected.

- *alpha, delta, u, i, class, redshift, plate, MJD, fieldbin, runbin*

Training Process:

- select k by accuracy.
- select model by misclassification rate



KNN Models - Misclassification, k =110

	GALAXY	QSO	STAR	average
1-fold	0.128	0.110	0.150	0.129
2-fold	0.132	0.113	0.163	0.136
3-fold	0.128	0.104	0.150	0.127
average	0.129	0.109	0.154	0.131

50% PI: 3-fold model has lowest misclassification rate

	GALAXY	QSO	STAR	average
1-fold	0.011	0.055	0.003	0.023
2-fold	0.009	0.057	0.002	0.023
3-fold	0.009	0.049	0.001	0.020
average	0.010	0.054	0.002	0.022

80% PI: 3-fold model has lowest misclassification rate

3-fold Model: on holdout set

	GALAXY	QSO	STAR	average
50% PI	0.133	0.104	0.150	0.129
80% PI	0.011	0.047	0.003	0.020
average	0.072	0.076	0.076	0.074

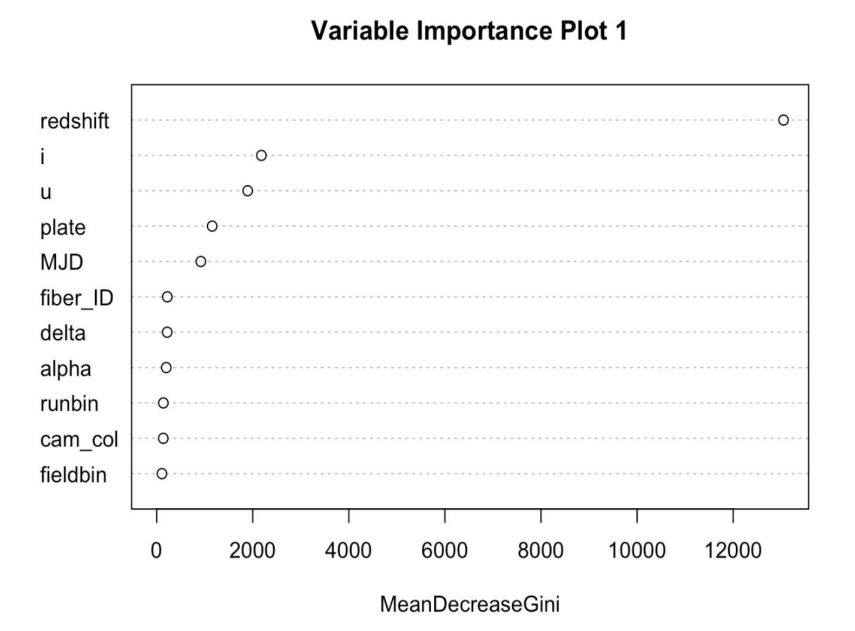
- The 80% prediction interval gives a smaller overall misclassification rate on the holdout set.
- The misclassification rate for GALAXY is the lowest
- NO overfitting, result similar to CV

KNN Model

Drawbacks in this project:

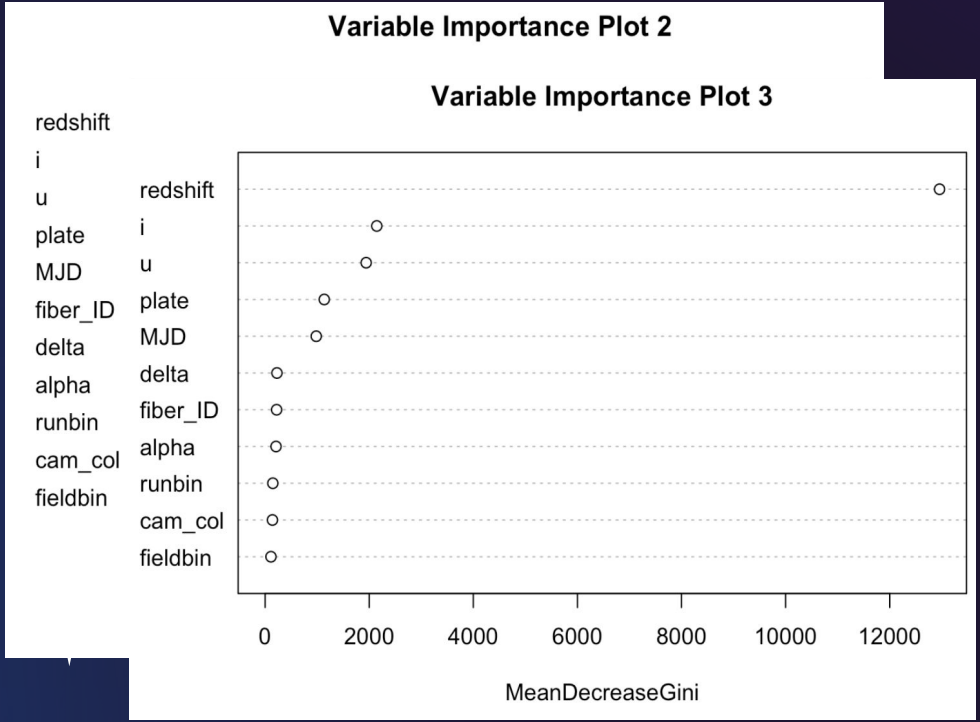
- long runtime ~ 20 min each fold (30,000 points) -> 60min for 3 folds!!!
- Euclidean Distance: hard to scale each variable
- Easy to Overfit (tone k bigger, 100-> 500?)
- Hard to interpret
- Statistical Models work much better!!!

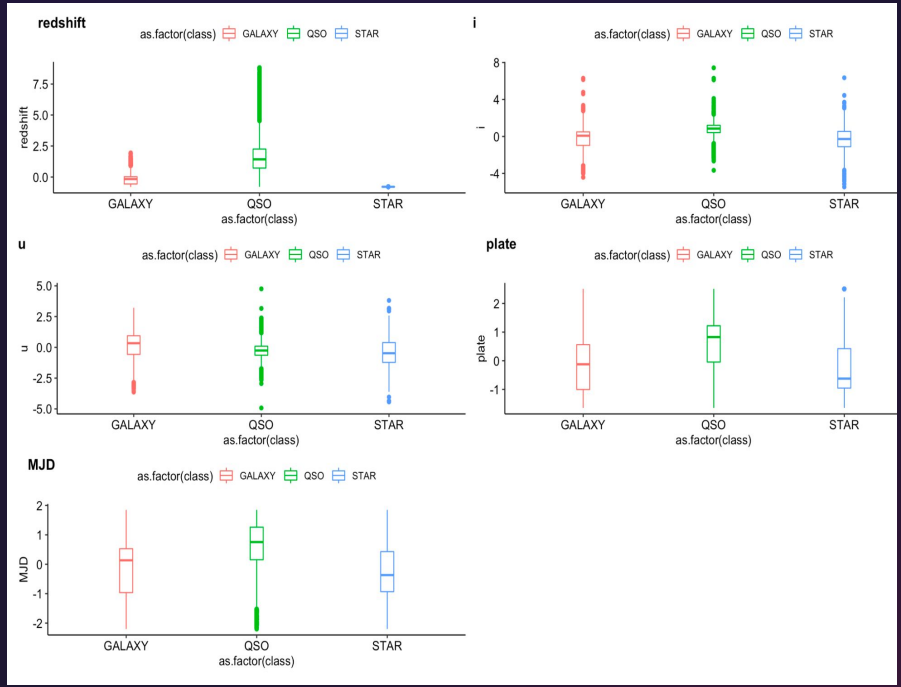
Random Forest



- library(randomForest)
- Number of trees: 400
- Prediction type: probability
- Use both 50% and 80% prediction interval
- 3-fold CV for undersampling: 30337 in train vs 15169 in test
- Top 5 most important variables: redshift, i,u,plate, MJD

RF model: variable importance plot





RF models - Prediction Interval

50% Prediction Interval Table for FOLD 1

	GALAXY	GALAXYQSO	GALAXYSTAR	QSO	QSOGALAXY	STAR
GALAXY	4808	7	1	146	5	37
QSO	274	4	0	4813	2	1
STAR	0	0	0	0	0	5070

80% Prediction Interval Table for FOLD 1

	GALAXY	GALAXYQSO	GALAXYSTAR	GALAXYSTARQSO	QSO	QSOGALAXY	STAR	STARGALAXY
GALAXY	4359	452	4	1	46	105	26	11
QSO	123	155	0	0	4541	274	1	0
STAR	0	0	0	0	0	0	5059	11

1-fold

50% Prediction Interval Table for FOLD 2

	GALAXY	GALAXYQSO	QSO	QSOGALAXY	STAR
GALAXY	4892	12	130	4	44
QSO	332	4	4689	5	0
STAR	0	0	0	0	5057

80% Prediction Interval Table for FOLD 2

	GALAXY	GALAXYQSO	GALAXYQSOSTAR	GALAXYSTAR	QSO	QSOGALAXY	STAR	STARGALAXY	STARQSO
GALAXY	4428	469	1	6	32	102	28	15	1
QSO	153	183	0	0	4437	257	0	0	0
STAR	0	0	0	0	0	0	5045	12	0

2-fold

50% Prediction Interval Table for FOLD 3

	GALAXY	GALAXYQSO	QSO	QSOGALAXY	STAR	STARGALAXY	STARQSO
GALAXY	4860	8	158	10	32	1	1
QSO	238	8	4752	10	3	0	0
STAR	0	0	0	0	5088	0	0

80% Prediction Interval Table for FOLD 3

	GALAXY	GALAXYQSO	GALAXYQSOSTAR	GALAXYSTAR	QSO	QSOGALAXY	STAR	STARGALAXY	STARGALAXYQSO	STARQSOGALAXY
GALAXY	4329	534	0	5	36	132	22	10	1	1
QSO	108	136	2	0	4483	279	1	2	0	0
STAR	0	0	0	0	0	0	5075	13	0	0

3-fold

RF Models - Misclassification Table

	GALAXY	QSO	STAR	average
1-fold	0.037	0.054	0	0.030
2-fold	0.034	0.066	0	0.033
3-fold	0.038	0.048	0	0.029
average	0.036	0.056	0	0.031

50% PI: 3-fold model has lowest misclassification rate

	GALAXY	QSO	STAR	average
1-fold	0.014	0.024	0	0.013
2-fold	0.012	0.030	0	0.014
3-fold	0.011	0.022	0	0.011
average	0.012	0.025	0	0.013

80% PI: 3-fold model has lowest misclassification rate

3-fold RF Model: on holdout set

	GALAXY	QSO	STAR	average
50% PI	0.038	0.049	0	0.029
80% PI	0.013	0.023	0	0.012
average	0.025	0.036	0	0.020

- The 80% prediction interval gives a smaller overall misclassification rate on the holdout set
- The misclassification rate for QSO is the highest, followed by GALAXY
- STAR has a misclassification rate equals to 0, which makes sense to us as the redshift is very distinguishable for STAR compare to the other two stellars.

Model Comparison : Misclassification Rate

50% prediction interval

	GALAXY	QSO	STAR
KNN	0.133	0.104	0.150
Multinomial	0.034	0.123	0.001
Random Forest	0.038	0.049	0.000

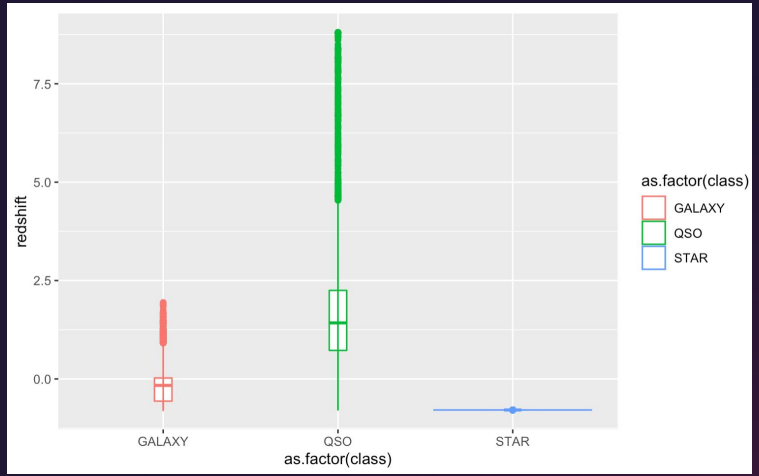
80% prediction interval

	GALAXY	QSO	STAR
KNN	0.011	0.047	0.003
Multinomial	0.015	0.071	0.000
Random Forest	0.013	0.023	0.000

Average Misclassification Rate

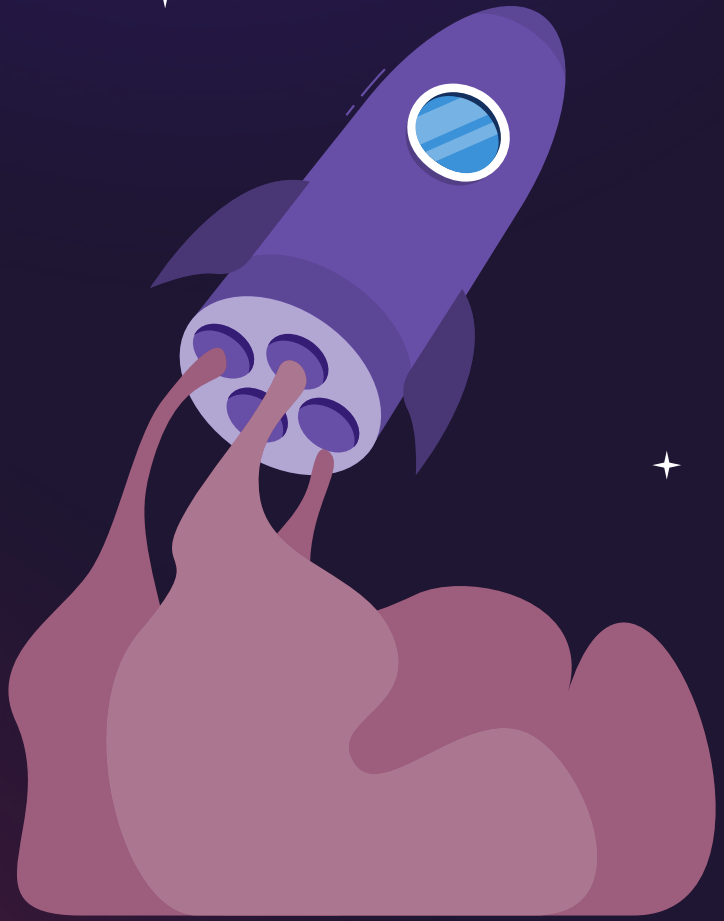
	GALAXY	QSO	STAR
KNN	0.072	0.076	0.076
Multinomial	0.025	0.097	0.000
Random Forest	**0.025**	**0.036**	**0.000**

Boxplot for redshift



Next Steps

- Avg length & interval score (self define)
- Keep KNN parameter toning
- Create more functions for code reproductivity.
- Code lint



THANKS!