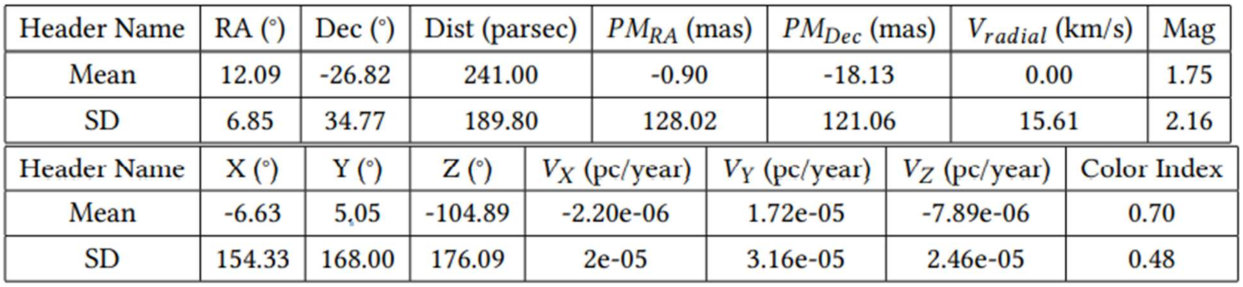
Stellar Classification and Clustering Analysis

STATS 780 – Final Project Report
Pranav Rawal—001316396
McMaster University

Introduction and Problem

Stellar classification is the procedure of classifying a star based on its spectral features. Stars can be composed of variable amounts of chemical elements, and the measured spectrum of those elements—termed spectral features—and other features such as color, luminosity, mass, and radius of the star can be used to classify the star into different classes. Stars are classified into the Morgan-Keenan system, a system grouping the temperature and luminosity of a star into 7 main classes (letters O, B, A, F, G, K, M) and 8 additional subclasses (Roman numerals from I to VII) if required. Current methods of star classification involve manual examination and verification—a time-consuming process [6]. The literature suggests automated methods utilizing knowledge-based systems [7], artificial neural networks [7], and statistical clustering techniques such as k-means [7], random forest [13][12][6]. The literature does not seem to have tested or analyzed methods seen in class, such as the Gaussian mixture model. In this analysis, star data obtained from the Hipparcos [9], Yale [10], and Gliese [11] star catalogs will be classified using the various clustering and classification techniques discussed in class. The catalogs are available in a combined dataset of 119614 stars for ease-of-access [8]. The combined dataset labels 25 fields, with 14 fields identifying stellar features such as right ascension, declination, distance, magnitude, and color index. All fields are real type, except for the categorical stellar classification. The data from the combined catalogs were filtered for complete cases over the 14 fields, and the total number of data points is 51084. A script written in Python 3.7.2 was used to reformat the original categorical data labels into the alphabetical and Roman numeral labels. A summary is seen below.

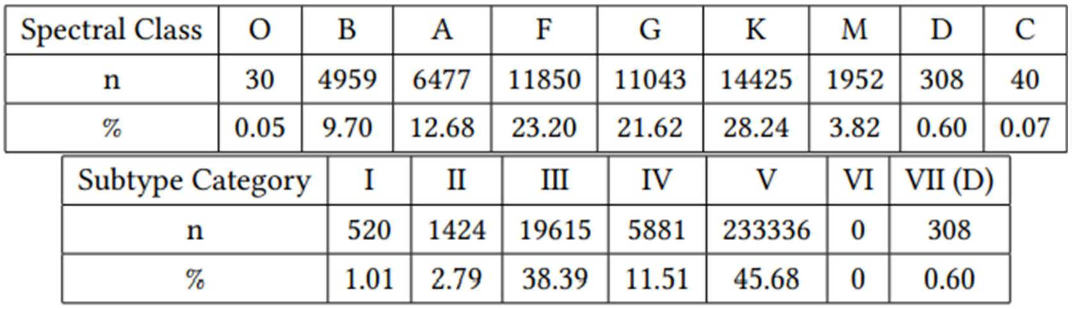
Table 1: Spectral Classification Data Summary - Real Variables



Header Name	RA (°)	Dec (°)	Dist (parsec)	PM_{RA} (mas)	PM_{Dec} (mas)	V_{radial} (km/s)	Mag
Mean	12.09	-26.82	241.00	-0.90	-18.13	0.00	1.75
SD	6.85	34.77	189.80	128.02	121.06	15.61	2.16

Header Name	X (°)	Y (°)	Z (°)	V_X (pc/year)	V_Y (pc/year)	V_Z (pc/year)	Color Index
Mean	-6.63	5.05	-104.89	-2.20e-06	1.72e-05	-7.89e-06	0.70
SD	154.33	168.00	176.09	2e-05	3.16e-05	2.46e-05	0.48

Table 2: Spectral Classification Data Summary - Categorical Variables



Spectral Class	O	B	A	F	G	K	M	D	C
n	30	4959	6477	11850	11043	14425	1952	308	40
%	0.05	9.70	12.68	23.20	21.62	28.24	3.82	0.60	0.07

Subtype Category	I	II	III	IV	V	VI	VII (D)
n	520	1424	19615	5881	233336	0	308
%	1.01	2.79	38.39	11.51	45.68	0	0.60

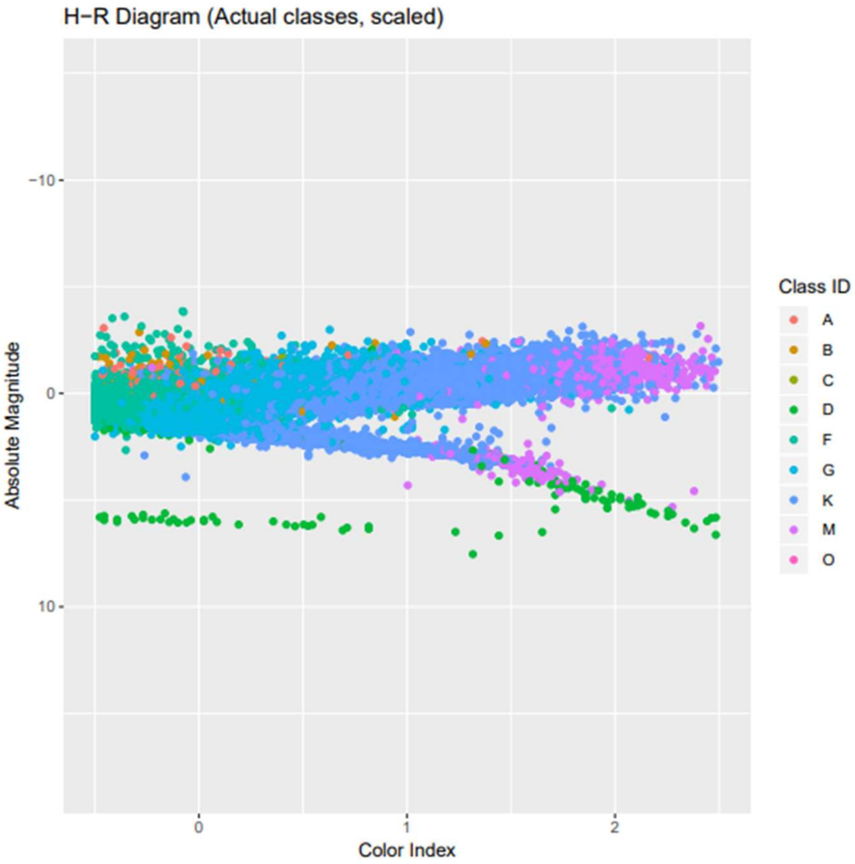
The goal of this paper is to determine if newer clustering and classification methods can provide results comparable to those recommended by literature. The literature recommends random forests, k-means, neural network methods as the usual techniques, however it does not mention any other techniques. These methods as well as Gaussian and non-Gaussian mixture models will be used to cluster and classify the star data. A comparison of the methods will be discussed. The literature usually depicts the data plotted as a chart called the "H-R diagram", with the color index of the star on the X-axis and absolute magnitude on the Y-axis. The Y-axis is plotted with an inverted axis as a historical convention. The clustering and classification analysis of the data is done using R - version 3.5.2. The randomForest - 4.6-14[4] package will be used for analysis using random forests, cluster - 2.0.7-1[1] package for k-means analysis, nnet - 2.0.7-1[1] package for neural networks, mclust - 5.4.2[2] package for Gaussian model-based clustering, and MixGHD - 2.3.1[5] for non-Gaussian model-based clustering.

Techniques and Parameters

A decision tree is a data mining method that predicts or classifies input data using a tree. The nodes of the tree act as branch points where input data is limited to follow a path to the branch terminus based on the value of that input. At the terminus of a branch, the input variable can be classified into a category dependent on the path taken. The random forest method uses a large collection of decision trees in order to determine the best variables to use as a decision at each node of the tree. For the random forest modelling method, the parameters $mtry$ and $ntree$ can be changed to alter the performance of the model. $mtry$ details the number of predictors to be considered at each branch of the decision tree. Only a subset of predictors are used in the random forest model, so $mtry$ must be less than the number of predictors. A common value for $mtry$ is$\sqrt{n}$, where n is the number of predictors. The parameter $ntree$ sets the number of trees with which to generate. Too many trees can lead to overfitting, while too little will lead to poor test error convergence. 2 K-means clustering is an unsupervised technique that uses the k-means algorithm in order to partition observations into k clusters. However, knowledge of k must be known first. To determine k, an elbow plot can be used–a plot of number of clusters versus within cluster sum of squares (increasing the amount of explained variance in the data as you increase k). The plot usually has a sharp decrease at the "elbow", and k is chosen at that value. A neural network is a non-linear statistical model that uses hidden weights of Z and linear combinations of X in order to classify or predict Y. The hidden unit Z usually has the form of Zm = σ ($\alpha_0 m$ + $\boldsymbol{\alpha}'_{m}$X), where σ is the sigmoid activation function, σ (ν) = 1/(1 – $e^{-\nu}$). The hidden units of Z are then further linearly combined to form T—the output, where Tk = $B_0 k$ + $\mathbf{B}'_{k}$Z. Parameters of neural networks include how many hidden layers to use $size$, as well as the $decay$ or sharpness of the sigmoid or activation function. To determine these parameters, k-fold cross validation technique can be used to determine the parameter values that give the least error. K-fold cross validation, is a technique that splits up data into k groups, usually five, where each group is used as training data for the algorithm. One group is kept left out as a validation set to test the model. This process is repeated and averaged for a final result. This allows each data point to be used for training and validation, but only used once for validation to prevent any bias. K-fold cross validation is used to select the best parameters for the neural network classifier. Mixture model-based clustering such as the Gaussian parsimonious clustering method and generalized hyperbolic mixture model fits sets—mixtures—of Gaussian distributions or hyperbolic distributions to the data set. These distributions are centered around the clusters, and mixtures of distributions can thus be used to cluster and classify data. The hyperbolic mixture model was chosen in this investigation because it is a generalization of the Gaussian distribution, skew-t distribution, asymmetric Laplace distribution, and others [5]. To select between the best mixture clustering models, the Bayesian information criterion (BIC) can be used. After each model is fit to the data, the model which provides the best performance must be determined. The Rand index is a metric that measures the similarity between data clusters, namely our validation set and the predicted class set. The adjusted Rand index (ARI) is a modification to the Rand index, correcting it for random chances, and higher values of ARI are better than lower ones. For this reason, the ARI to rank the performance of each of the classification techniques.

Results and Conclusion

A split of 75% (n = 38313) for training and 25% (n = 12771) for validation was used. A plot of the test set stellar data with known/correct labels is shown below.



Random Forest

For random forests, the default parameters of *mtry* = 1 and *ntree* = 500 provided good results. *mtry* was tested for bales from 1 to 3, and *ntree* was tested with values 100 to 500. The test error graph is not shown as the data set was too large for the plot to be produced in a reasonable amount of time due to hardware limitation of the computer. Random forests were generated using the 75% split of test data, and then the remaining 25% of validation data was classified and compared with the known spectral class. The ARI for the validation set suing random forests was 0.628. The classified validation set using random forests is shown on the next page.

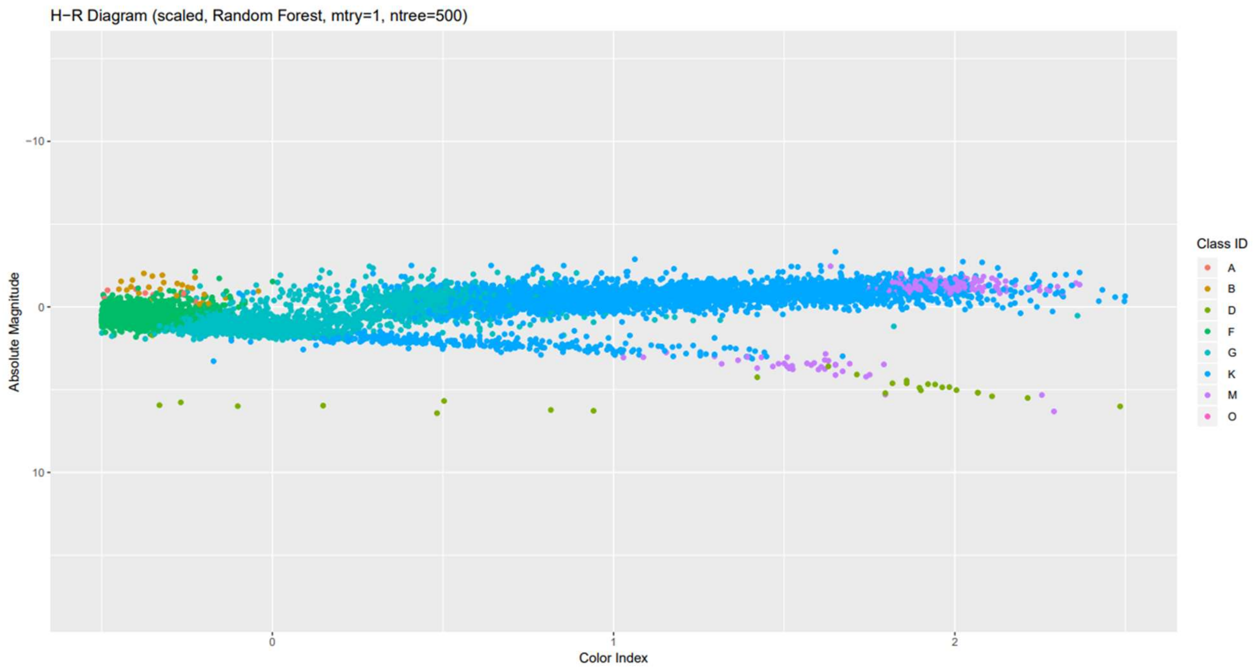
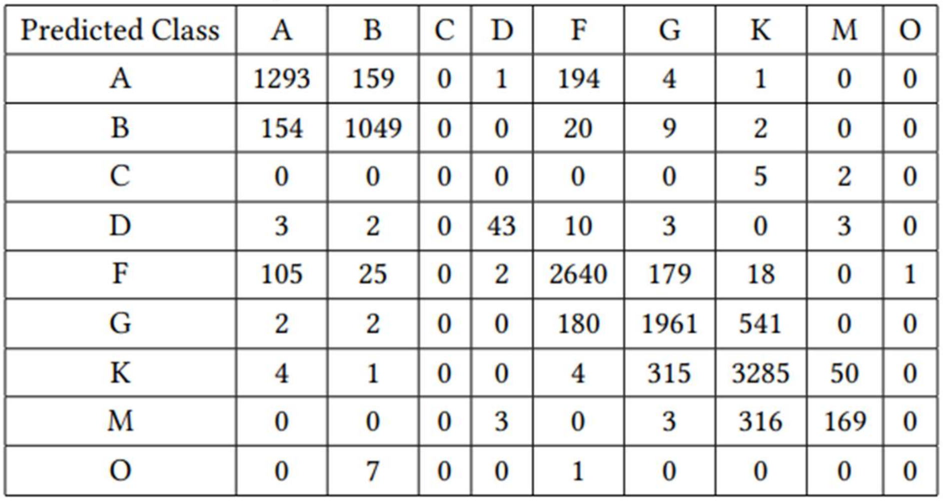


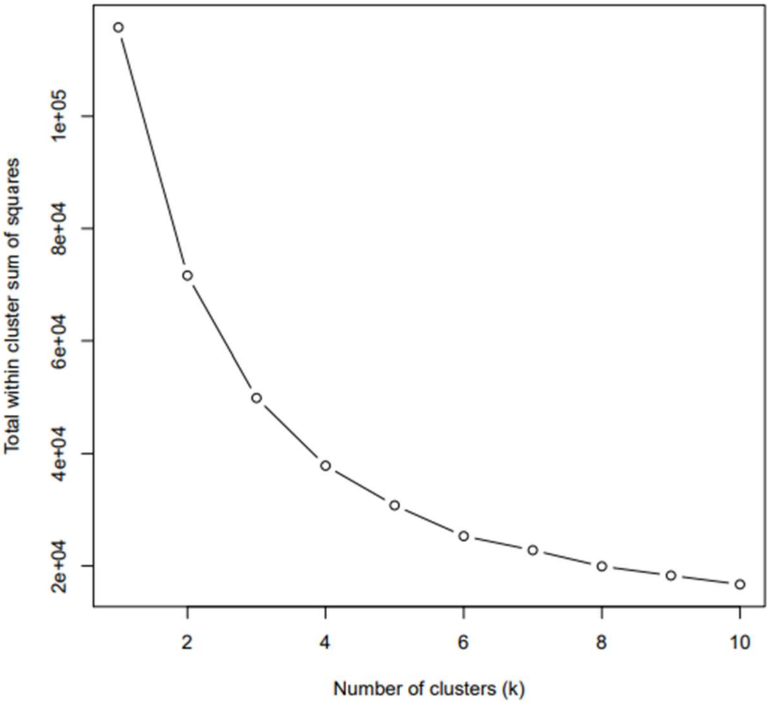
Table 3: Spectral Classification - Random Forests



Predicted Class	A	B	C	D	F	G	K	M	O
A	1293	159	0	1	194	4	1	0	0
B	154	1049	0	0	20	9	2	0	0
C	0	0	0	0	0	0	5	2	0
D	3	2	0	43	10	3	0	3	0
F	105	25	0	2	2640	179	18	0	1
G	2	2	0	0	180	1961	541	0	0
K	4	1	0	0	4	315	3285	50	0
M	0	0	0	3	0	3	316	169	0
O	0	7	0	0	1	0	0	0	0

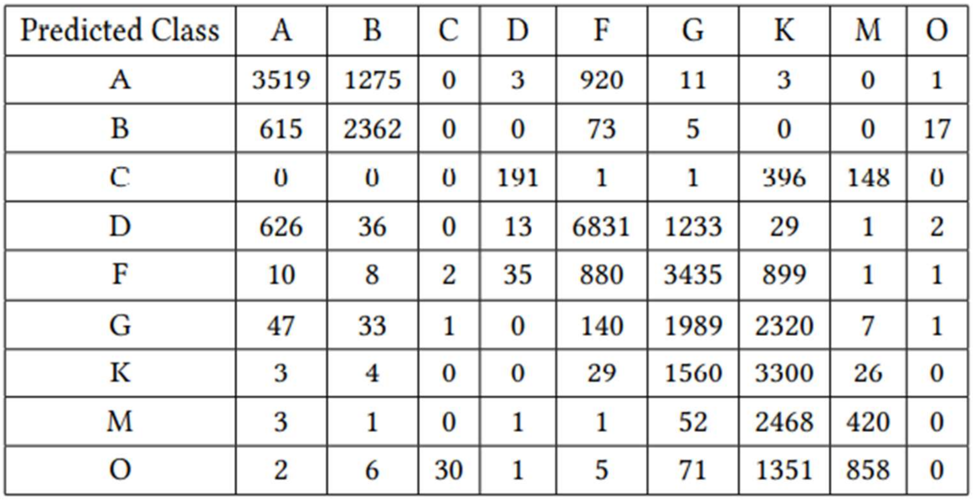
K-Means Clustering

For determining which k value to use to partition the observation data in k-means, an elbow plot was used. The elbow plot does not show a clear value of which cluster to use, indicative of the non-clustering of the training data.

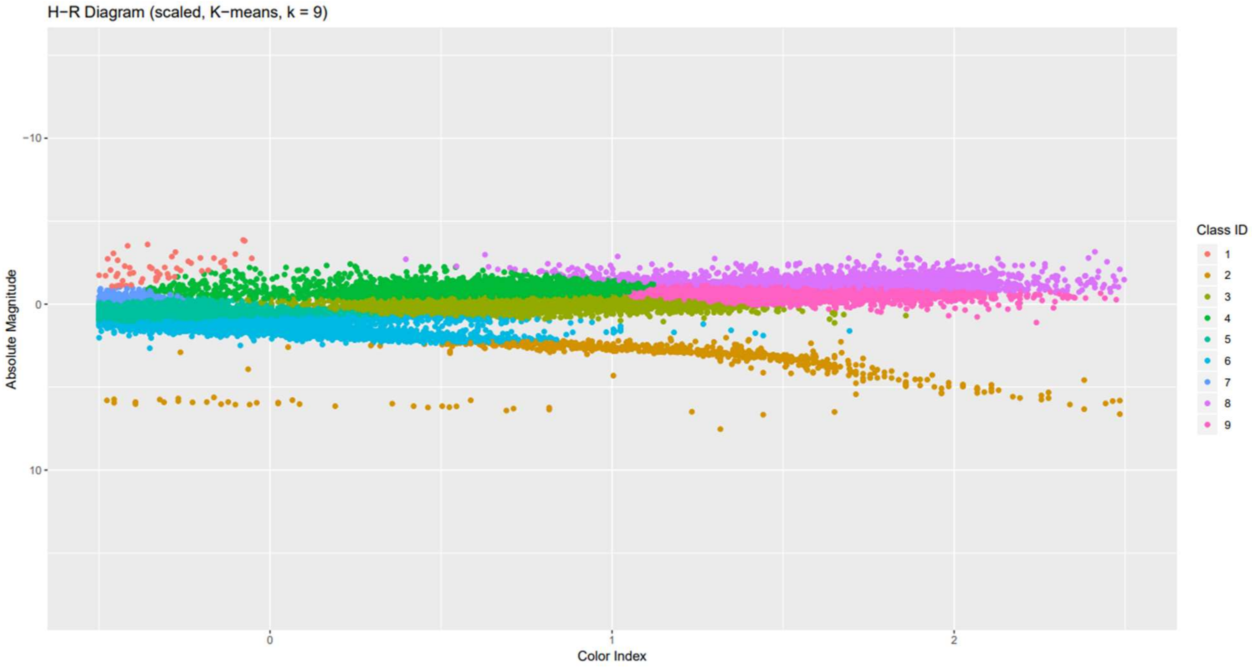


However, the number of expected classes for this particular dataset is known, thus setting k to 9 and using the k-means algorithm yielded the following results. In this manner, the k-means algorithm will fit nine unlabelled clusters to the data. The ARI of the 25% validation set was 0.33. A plot of the k-means classifier output is shown on the next page.

Table 4: Spectral Classification - K-means

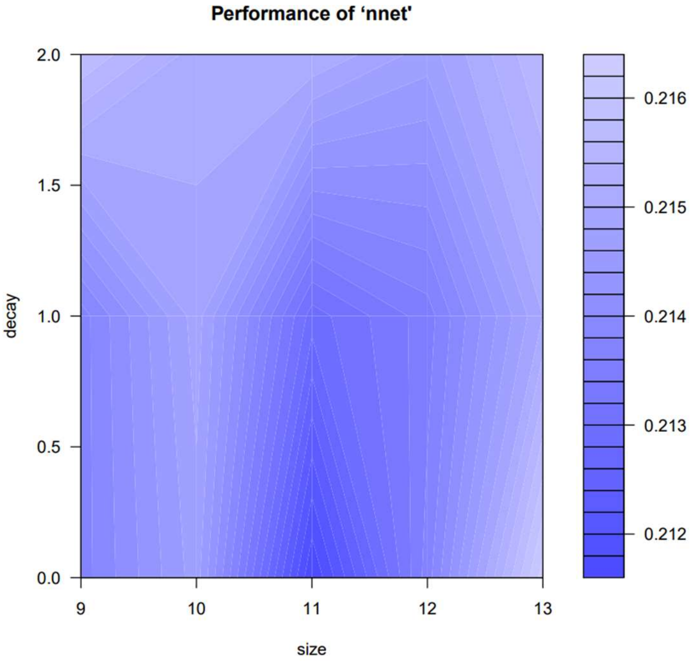


Predicted Class	A	B	C	D	F	G	K	M	O
A	3519	1275	0	3	920	11	3	0	1
B	615	2362	0	0	73	5	0	0	17
C	0	0	0	191	1	1	396	148	0
D	626	36	0	13	6831	1233	29	1	2
F	10	8	2	35	880	3435	899	1	1
G	47	33	1	0	140	1989	2320	7	1
K	3	4	0	0	29	1560	3300	26	0
M	3	1	0	1	1	52	2468	420	0
O	2	6	30	1	5	71	1351	858	0



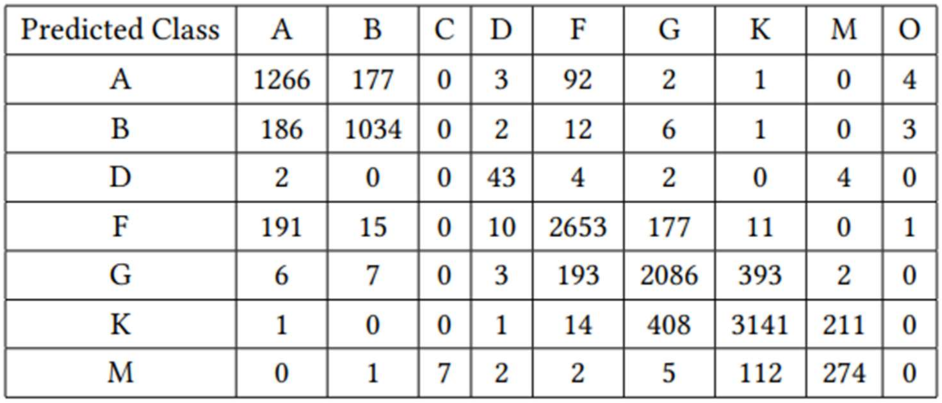
Neural Networks

For the neural network, the k-fold cross validation technique was used to determine the most suitable values for *size* and *decay* parameters. The technique searched a space of size from 1 to 30 and a *decay* from 0 to 5. A zoomed plot of the least error is shown int the plot below. The final values of 11 and 0 were chosen for *size* and *decay* respectively.

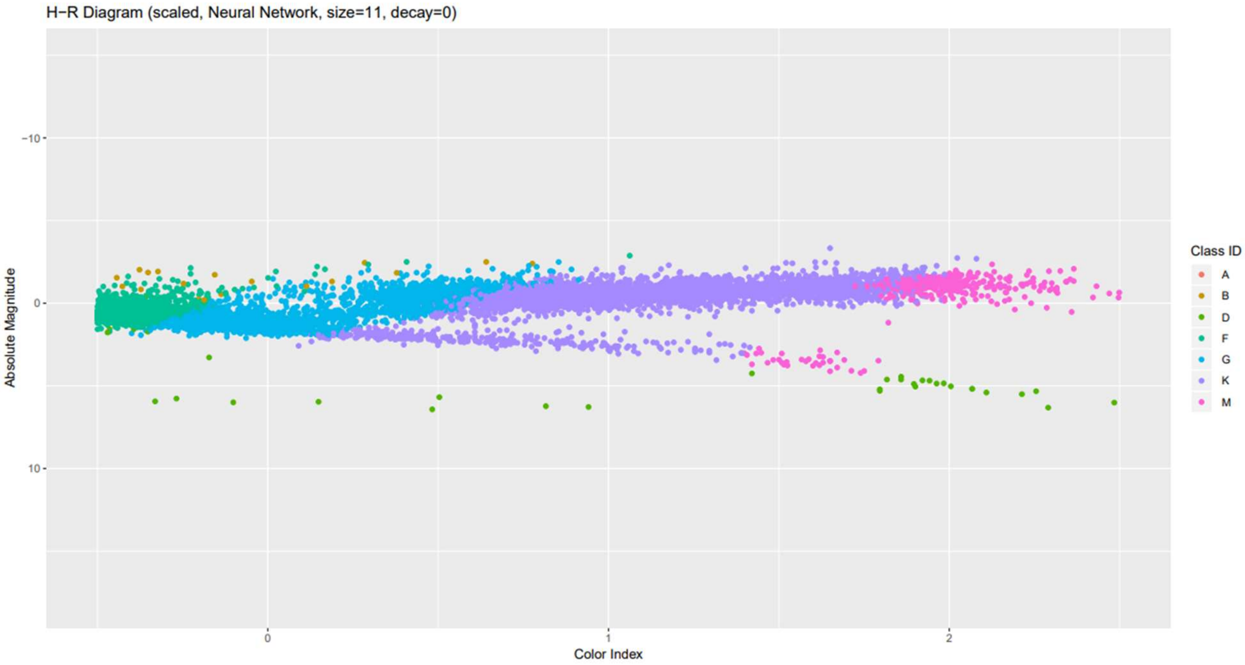


The ARI of the neural network model when applied to the 25% validation set was 0.63. A plot of the output of the classifier is shown on the next page.

Table 5: Spectral Classification - Neural Network

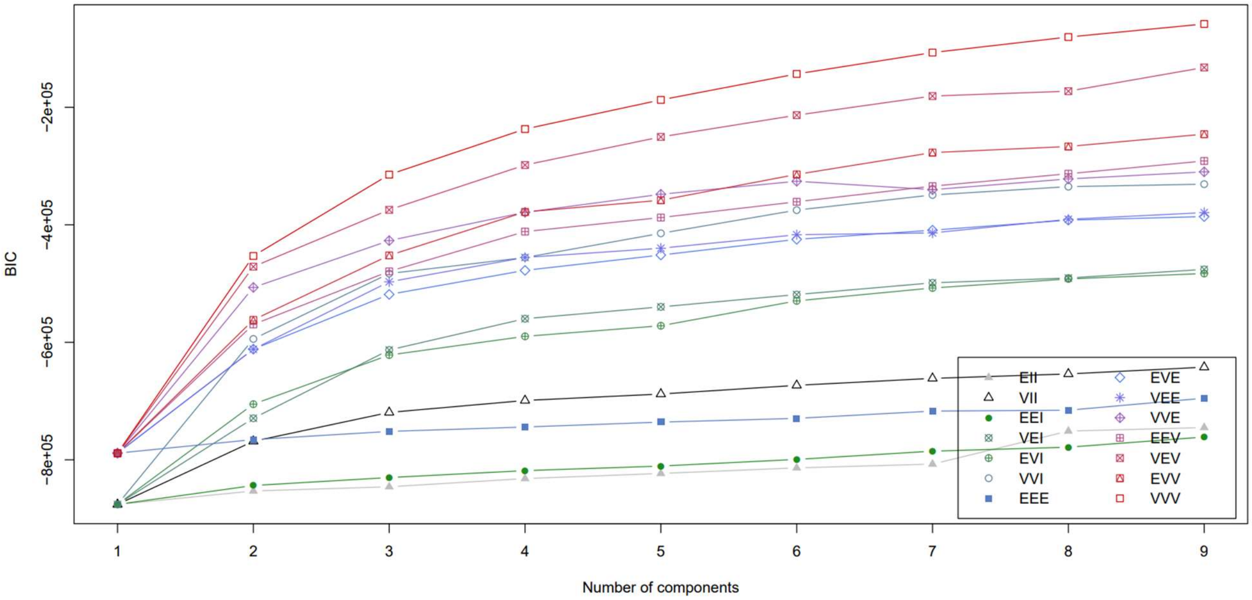


Predicted Class	A	B	C	D	F	G	K	M	O
A	1266	177	0	3	92	2	1	0	4
B	186	1034	0	2	12	6	1	0	3
D	2	0	0	43	4	2	0	4	0
F	191	15	0	10	2653	177	11	0	1
G	6	7	0	3	193	2086	393	2	0
K	1	0	0	1	14	408	3141	211	0
M	0	1	7	2	2	5	112	274	0



Model-based clustering

For both Gaussian and non-Gaussian mixture models, the classification could only be performed on the training set. The software was unable to predict using the validation set. This may be an issue in the libraries, or the input data may be malformed in some way. The following results are given in terms of the training set. For Gaussian mixture models, the top models were VVV, VEV, and EVV as seen in the BIC plot below. VII was chosen as the mixture model for classifying the training set.



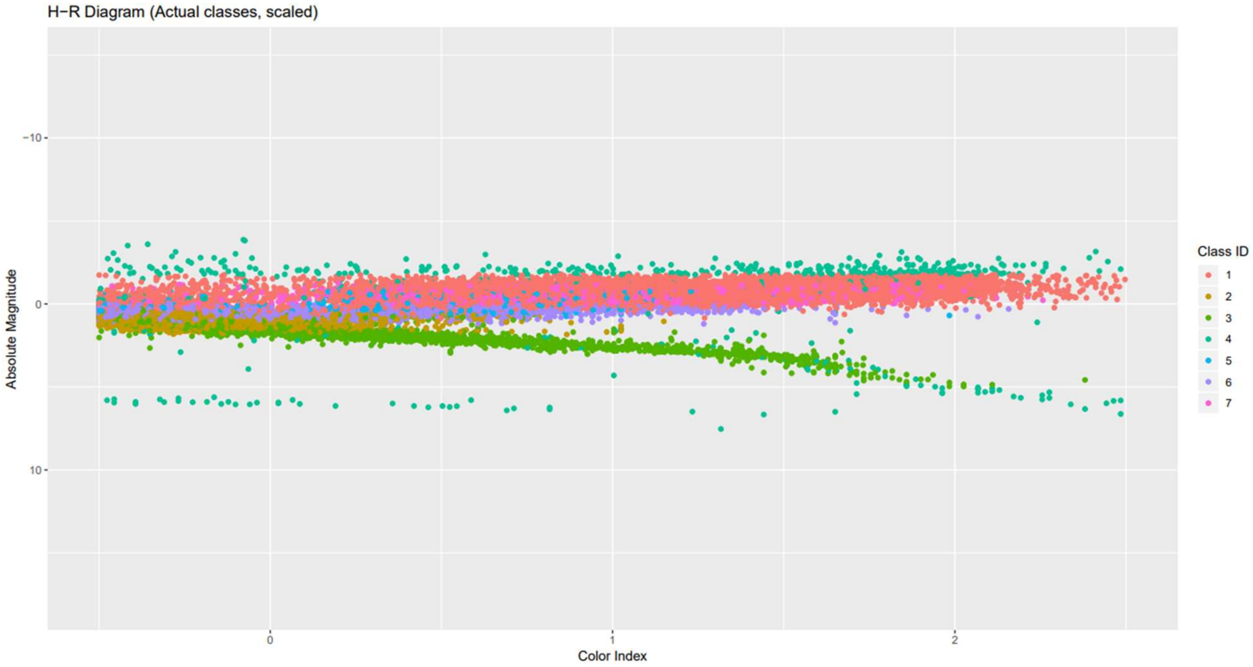
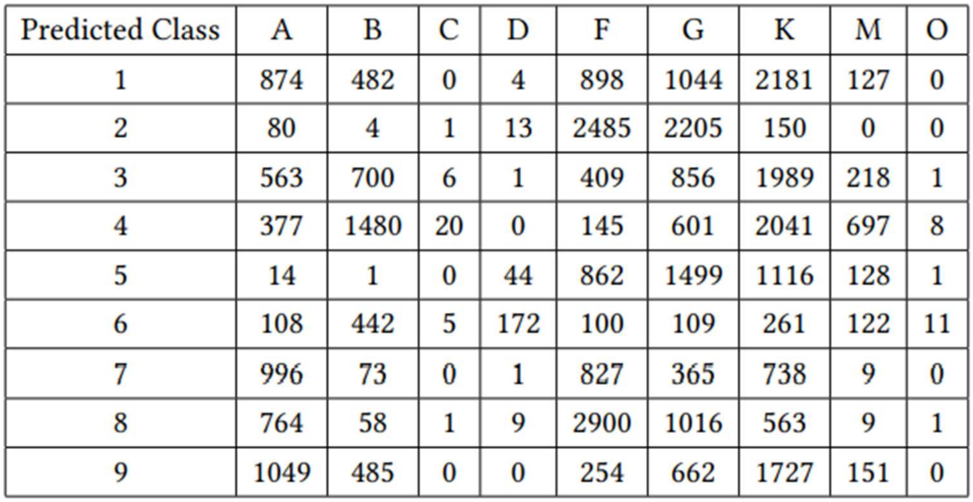


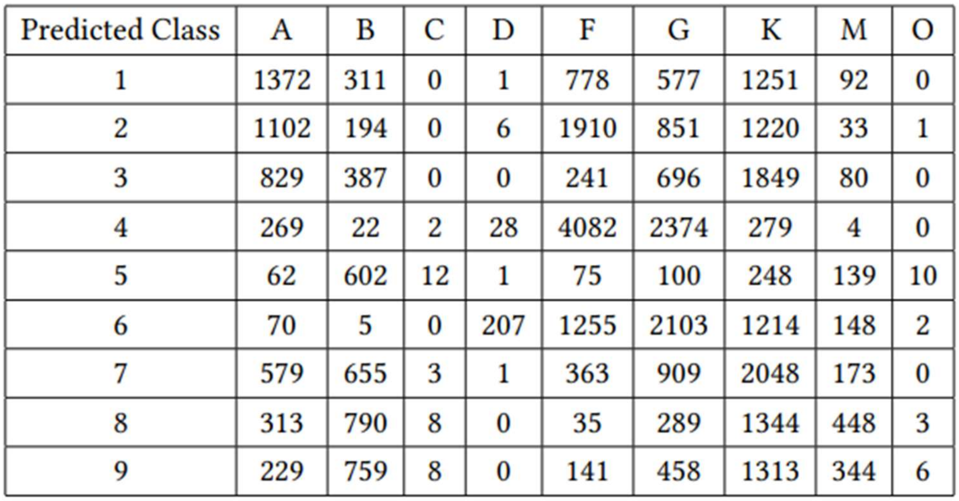
Table 6: Spectral Classification – Generalized Gaussian Distribution



Predicted Class	A	B	C	D	F	G	K	M	O
1	874	482	0	4	898	1044	2181	127	0
2	80	4	1	13	2485	2205	150	0	0
3	563	700	6	1	409	856	1989	218	1
4	377	1480	20	0	145	601	2041	697	8
5	14	1	0	44	862	1499	1116	128	1
6	108	442	5	172	100	109	261	122	11
7	996	73	0	1	827	365	738	9	0
8	764	58	1	9	2900	1016	563	9	1
9	1049	485	0	0	254	662	1727	151	0

The classified training data is seen below for the generalized hyperbolic distribution models. Interpretation of the model was difficult, with sparse documentation.

Table 7: Spectral Classification - Generalized Hyperbolic Distribution

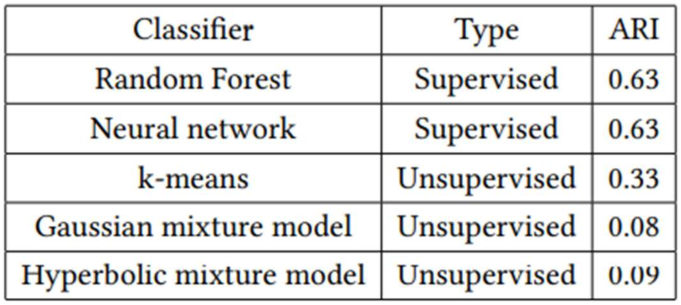


Predicted Class	A	B	C	D	F	G	K	M	O
1	1372	311	0	1	778	577	1251	92	0
2	1102	194	0	6	1910	851	1220	33	1
3	829	387	0	0	241	696	1849	80	0
4	269	22	2	28	4082	2374	279	4	0
5	62	602	12	1	75	100	248	139	10
6	70	5	0	207	1255	2103	1214	148	2
7	579	655	3	1	363	909	2048	173	0
8	313	790	8	0	35	289	1344	448	3
9	229	759	8	0	141	458	1313	344	6

The ARI for Gaussian mixture modelling was 0.08, while the ARI for generalized hyperbolic distribution mixture modelling was 0.09.

Conclusion

Table 8: Comparison of Classification Techniques



Classifier	Type	ARI
Random Forest	Supervised	0.63
Neural network	Supervised	0.63
k-means	Unsupervised	0.33
Gaussian mixture model	Unsupervised	0.08
Hyperbolic mixture model	Unsupervised	0.09

In conclusion, the unsupervised models, specifically k-means, Gaussian mixture, and non-Gaussian hyperbolic distribution mixture modelling algorithms performed the worst when compared to the supervised models of neural networks and random forests. Random forests and neural networks performed equally well, returning an ARI of 0.63 for the 25% validation set. The unsupervised algorithms performed very poorly, with the mixture-model based returning a very low ARI close to 0. This means that the performance of mixture models is no better than just randomly clustering the data. Between neural networks and random forests, it is interesting to note that the neural network technique was not able to distinguish either C-type or O-type stars.

In terms of the data, C-type, O-type and D-type stars were the most unique stars and this may be the reason why the neural network was unable to distinguish them. C-type stars are not part of the standard Morgan-Keenan system, and emit a different spectra than most other stars. Their luminosity and color however, places them nearer to M or K-type stars, leading to their misclassification by the random forest as a member of those classes. O-type stars are extremely short-lived, and will quickly degenerate into another class of star. Additionally, because O-type, C-type, and D-type are among the rarest types of stars, the total counts of 378 of them among the 50 thousand stars in this dataset may have not enough samples of them to provide a proper training set. A more balanced data set or stratified sampling to even out the relative amounts of each class may lead to better results.

References

[1] Maechler, M., Rousseeuw, P., Struyf, A., Hubert, M., Hornik, K.(2018). cluster: Cluster Analysis Basics and Extensions. R package version 2.0.7-1. https://CRAN.R-project.org/package=cluster

[2] Scrucca L., Fop M., Murphy T. B. and Raftery A. E. (2016). *mclust 5: clustering, classification and density estimation using Gaussian finite mixture models.* The R Journal 8/1, pp. 205-233 https://cran.r-project.org/package=mclust

[3] Venables, W. N. and Ripley, B. D. (2002) *nnet: Feed-Forward Neural Networks and Multinomial Log-Linear Models.* R package version 7.3-12. https://CRAN.R-project.org/package=nnet

[4] A. Liaw and M. Wiener (2002). *Classification and Regression by randomForest - version 4.6-14.* R News 2(3), 18–22. https://cran.rproject.org/package=randomForest

[5] Tortora, C. et al. (2015). *MixGHD: Model Based Clustering, Classification and Discriminant Analysis Using the Mixture of Generalized Hyperbolic Distributions.* R package version 2.3.1 https://cran.r-project.org/package=MixGHD

[6] Carricajo, I. and Manteiga, M. (2004). *Automatic Classification of Stellar Spectra.* Lecture Note and Essays in Astrophysics, pp. 153-164 http://adsabs.harvard.edu/abs/2004LNEA....1..153C

[7] Dafonte, C. et al. (2005). *A Comparative Study of KBS, ANN and Statistical Clustering Techniques for Unattended Stellar Classification.* CIARP 2005 Proceedings of the 10th Iberoamerican Congress Conference on Progress in Pattern Recognition, Image Analysis and Applications, pp. 566-577 http://dx.doi.org/10.1007/11578079_59

[8] David Nash, (2006*). Astronexus HYG Star Database Version 3* https://github.com/astronexus/HYG-Database

[9] ESA, 1997, The Hipparcos and Tycho Catalogues, ESA SP-1200 https://www.cosmos.esa.int/web/hipparcos/catalogues

[10] Hoffleit, D. and Warren, Jr., W.H., (1991), *The Bright Star Catalog, 5th Revised Edition (Preliminary Version*). http://tdcwww.harvard.edu/catalogs/bsc5.html

[11] Gliese, W.; Jahreiß, H. (1991). *Preliminary Version of the Third Catalogue of Nearby Stars*. http://adsabs.harvard.edu/abs/1991adc..rept.....G

[12] Li, K., Lin, Y., and Qiu, K. (2019), *Stellar Spectra Classification and Feature evaluation Based on Random Forest*. arXiv:1903.07939 [astroph.IM]

[13] Bai, Y., Liu, J., and Wang, S. (2018), *Machine Learning Classification of Gaia Data Release 2*. arXiv:1808.05728 [astro-ph.SR]