STATS 4CI3
Final Project

Chris Stavnitzky 400091700

04/19/2021

Problem to Address

The dataset of interest is the Glass Identification Data Set from the USA Forensic Science Service (German 1987). This dataset consists of glass from different sources and physical properties of the glasses, including chemical composition and refractive index. The purpose of the dataset was to classify the type of glass based on its physical properties so it may be identified and then used as evidence in court.

Description of Data

The glasses are sourced from building windows and vehicle windows, containers, tableware, and headlamps.

These sources are split into a total of seven classes: float processed window glass, non-float processed window glass, float processed vehicle glass, non-float processed vehicle glass, containers, tableware, headlamps.

Despite being described by the collectors of the dataset, there is no non-float processed vehicle glass in the 214 observations of this dataset.

Float glass is a sheet of glass made by floating molten glass on a bed of molten metal. This method gives the sheet uniform thickness and very flat surfaces.

The physical characteristics which are suspected to be useful in identification of the type of glass and those which will be used as predictor variables for classification in this dataset include:

refractive index, sodium content, magnesium content, aluminium content, silicon content, potassium content, calcium content, barium content, iron content.

In the following table we have compiled summary statistics for these predictor variables

Variable	Min	p25	Mean	Median	p75	Max	Standard_Deviation	iqr
RI	1.51	1.52	1.52	1.52	1.52	1.53	0.00	0.00
Na	10.73	12.91	13.41	13.30	13.83	17.38	0.82	0.92
Mg	0.00	2.11	2.68	3.48	3.60	4.49	1.44	1.49
Al	0.29	1.19	1.44	1.36	1.63	3.50	0.50	0.44
Si	69.81	72.28	72.65	72.79	73.09	75.41	0.77	0.81
K	0.00	0.12	0.50	0.56	0.61	6.21	0.65	0.49
Ca	5.43	8.24	8.96	8.60	9.17	16.19	1.42	0.93
Ba	0.00	0.00	0.18	0.00	0.00	3.15	0.50	0.00
Fe	0.00	0.00	0.06	0.00	0.10	0.51	0.10	0.10

As the statistics show, the barium and iron content is quite low and the silicon and sodium content is quite high compared to the concentrations of the other elements. The units of the reflective index is similar to most of the units of concentration of the elements. Scaling of the variables will ensure these numerical differences in the predictor variables don't affect clustering.
Below is a parallel coord. plot which includes boxplots of the 6 different types of glass in the charted over the 10 predictor variables.

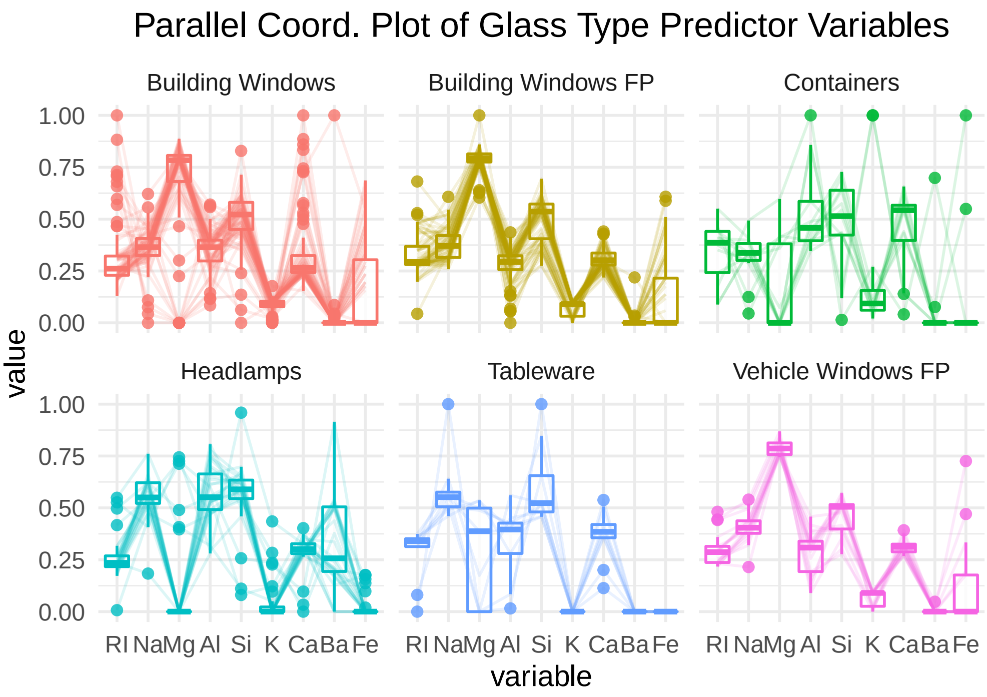


Figure 1: Parallel Coord. plot of the predictor variables. Each type has been scaled according to the minimum and maximum of the type.

From this plot we can see that the glasses seem to divide themselves well along the lines of certain chemicals such as the iron content, as seen in difference in iron in tableware glass compared to either types of building window glass, as well as the stark lack of magnesium content in headlamps compared to both types of building windows and the floated vehicle glass.

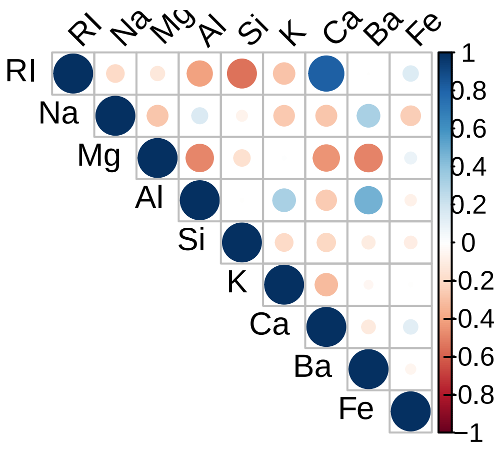


Figure 2: Correlation matrix of the chemical compositions and refractive index

Notably from this correlation plot we see that there is a very strong correlation between calcium content and refractive index, but as there aren't other strong positive correlations and only small to moderate correlations amongst the other variables it does not appear that additional measures to control the correlation among variables will have to be used.

Description of techniques used

Random Forest

The first technique we used to examine the data was an ensemble technique called random forest. Random forest is a form of classification tree. A classification tree is sort of flow chart or diagram which is representative of a series of decisions made regarding observations of a item which result in classifying that item into a particular category. An intuitive example of this process applied to the glass under discussion would be, based on the parallel coordinate plot, “Does the glass under inspection have an iron content > 0?” If it does, it is unlikely to be of the category of tableware glass, but more likely to be of the category of building window glass. By making effective and insightful thresholds to the question criteria to “branch” into the different categories, one can reach the end of the tree, the “leaves,” with accurate categorizations of the items.

A problem arises in this method where as one increases the number of “branches” or decisions made to categorize the item, the tree will tend to fixate on highly irregular patterns that are likely only present in one’s current data set, which leads to overfitting. This overfitting means the tree has very high variance but little bias. A remedy for this issue comes about through the process of bagging, which is a process of using a bootstrapping resampling technique to reduce the variance. This process is as follows: First, sample with replacement n sets of predictor/response variables and then train a classification tree on this selected data. Then, repeat this process B times to train B seperate classification trees. Finally, to classify a new item, get all B trees to classify it, and then average the result between all the trees. This averaging reduces the variance of the classification compared to a single tree, because as each tree is trained on essentially different training data, an average of all their classifications smooths the result to minimize the contribution of how overfitted each tree is to the data to the final classification.

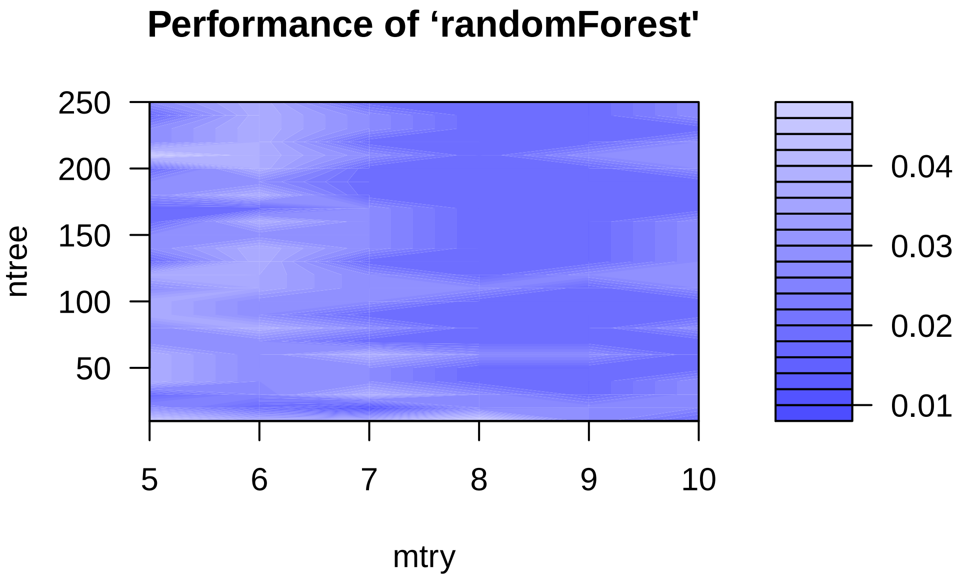
The remedy of bagging however causes another issue which needs solving. If some of the predictor variables are noticeably more useful in prediction then others, then during bagging most of the trees will use those variables for decisions in order to classify. This will cause correlation between the B trees which partially undos the variance reduction of the averaging. In order to minimize this correlation between trees, one randomly selects a subset of the predictor variables when training each tree, and only uses the members of this subset to create the resulting tree. This final modification leads to random forest.

If there are p predictor variables then usually for classification a subset of $\lfloor \sqrt{p} \rfloor$ variables are selected to train each tree, however this parameter can and most likely should be tuned based on each problem. The B trees trained is a free parameter and is optimized usually using cross-validation.

Cross-validation is a resampling technique which aims to test how well a statistical model will generalize to data beyond that currently available. K-fold cross-validation is a version of this technique. K-fold cross-validation consists of splitting one’s data into k equal sized groups, training one’s model on $k - 1$ of the groups, and using the remaining group to validate this model by seeing how successfully it classifies the members of the group. This process is then repeated so that each group is used precisely once as a validation group. One can then average the k results provided by each round to give a single judgment on classification.

For my application of random forest to classify the glass data; first I split the data into a 50/50 test and train set. Then I scaled the 10 predictor variables. Next I did 25 fold cross validation on a random forest whose number of predictor variables chosen per tree ranged from 5 to 10. From the initial analysis of the data descriptively, I felt that inclusion of any less than 5 of the predictor variables would seriously limit the power of the classification, so I chose to test 5 variables to all of them. The 25 different tree sizes from the cross-validation started at ntree = 10 and increasing the size by 10 until a maximum size of 250. This selection was primarily due to how broad it is and how not only does it cover a wide range of tree size, but also the small step size of 10 provides precise information as to where tree sizes are effective. This random forest was done using the R randomforest library. (Leo Breiman, Adele Cutler, and Wiener 2018-03-22)

The following is a plot showing the results of this cross validation as the prediction errors of random forests with the number of trees on the vertical axis and the number of predictor variables selected per tree trained on the horizontal axis. Darker blue means lower prediction error rate.



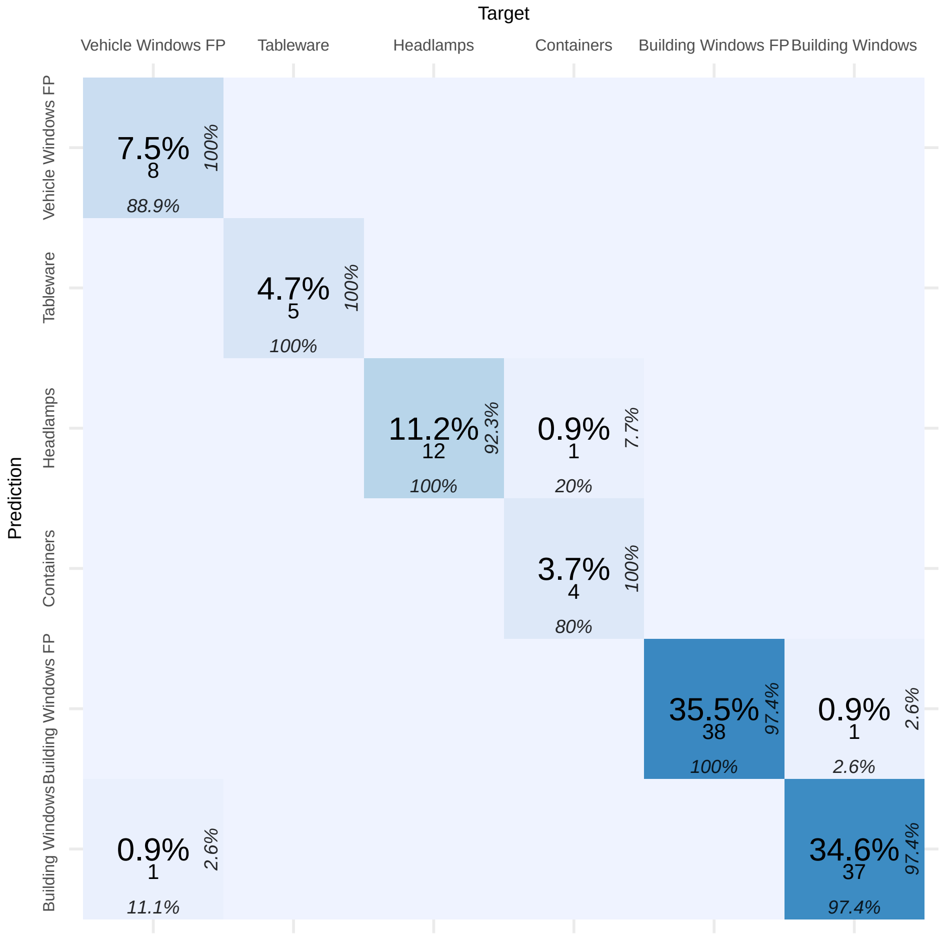
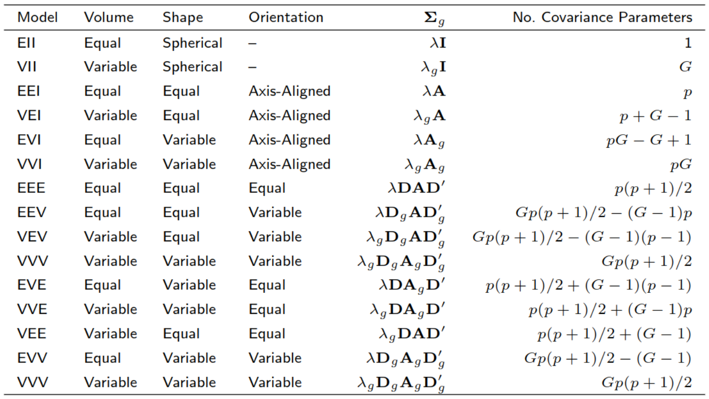


Figure 3: Confusion matrix from the final random forest classification of the test set. In the center of each tile is the overall percentage of classified items, beneath it the count. At the bottom is the fraction of glass of a given type that were predicted to be that type, and at the right is the fraction of predictions of a given type that actuallly were that given type.

This cross validation resulted in a selection of a random forest using 7 variable sized subsets and tree size of 40. The final classification error this tree had on the testing set was 0.02803738. The adjusted rand index of this model was 0.9395585.

The EM Algorithm

The second technique used to examine the glass dataset was a Gaussian parsimonious clustering model (GPCM) with variance structure trained via the EM algorithm. A mixture model is a statistical model based on a probability distribution which is created out of the weighted sum of multiple different probability distributions. The weighting is given as a total fraction and thus the weights add to one. This kind of model is useful in a clustering context when one believes that there are different groups in the data which are distributed differently. A gaussian mixture model (GMM) is a mixture model where the mixed probability distributions are normal distributions with different parameters. A GPCM is a gaussian mixture model where the covariance structure between the normal distributions is given one or more constraints which are ideally supposed to better adapt the GMM to the structure of the data for the problem at hand. The following is a chart of the most popular family of constraints given to the covariance matrices of the distributions in the model. Volume, shape, and orientation are what happens to the volume, shape, and orientation of all the distributions in the model under each different covariance constraint. The covariance matrices in the table are represented by the eigen-decomposition $\mathbf{\Sigma}_g = \lambda_g \mathbf{D}_g \mathbf{A}_g \mathbf{D}_g'$.



Model	Volume	Shape	Orientation	$\mathbf{\Sigma}_g$	No. Covariance Parameters
EII	Equal	Spherical	–	$\lambda\mathbf{I}$	1
VII	Variable	Spherical	–	$\lambda_g\mathbf{I}$	G
EEI	Equal	Equal	Axis-Aligned	$\lambda\mathbf{A}$	p
VEI	Variable	Equal	Axis-Aligned	$\lambda_g\mathbf{A}$	$p+G-1$
EVI	Equal	Variable	Axis-Aligned	$\lambda\mathbf{A}_g$	$pG-G+1$
VVI	Variable	Variable	Axis-Aligned	$\lambda_g\mathbf{A}_g$	pG
EEE	Equal	Equal	Equal	$\lambda\mathbf{DAD}'$	$p(p+1)/2$
EEV	Equal	Equal	Variable	$\lambda\mathbf{D}_g\mathbf{AD}_g'$	$Gp(p+1)/2-(G-1)p$
VEV	Variable	Equal	Variable	$\lambda_g\mathbf{D}_g\mathbf{AD}_g'$	$Gp(p+1)/2-(G-1)(p-1)$
VVV	Variable	Variable	Variable	$\lambda_g\mathbf{D}_g\mathbf{A}_g\mathbf{D}_g'$	$Gp(p+1)/2$
EVE	Equal	Variable	Equal	$\lambda\mathbf{DA}_g\mathbf{D}'$	$p(p+1)/2+(G-1)(p-1)$
VVE	Variable	Variable	Equal	$\lambda_g\mathbf{DA}_g\mathbf{D}'$	$p(p+1)/2+(G-1)p$
VEE	Variable	Equal	Equal	$\lambda_g\mathbf{DAD}'$	$p(p+1)/2+(G-1)$
EVV	Equal	Variable	Variable	$\lambda\mathbf{D}_g\mathbf{A}_g\mathbf{D}_g'$	$Gp(p+1)/2-(G-1)$
VVV	Variable	Variable	Variable	$\lambda_g\mathbf{D}_g\mathbf{A}_g\mathbf{D}_g'$	$Gp(p+1)/2$

(S. McNicholas 2021)

The EM algorithm is an iterative technique that can be used in situations where there is missing data amongst the observations of one's dataset or when it is possible to assume there is additional unknown/missing data points, sometimes referred to as "latent variables." In this case of use with a GPMM, one assumes there is missing data in the form of an indicator of membership of the population of one of the mixed probability distributions. One starts with the complete data log-likelihood

$$l_c(\vartheta) = \sum_{i=1}^{n} \sum_{g=1}^{G} z_{ig}[log(\pi_g) + log(\phi(\mathbf{x_i}|\mu_g, \mathbf{\Sigma}_g)]$$

Where z_{ig} is the indicator function of observation $\mathbf{x_i}$ being in component g where there are g different distributions in the mixture model. π_g is the weight of the g^{th} distribution, and ϕ is the density function of a multivariate normal distribution with mean $\mu_\mathbf{g}$ and covariance matrix $\mathbf{\Sigma_g}$. ϑ denotes the model parameters, which is the mixture weights, means, and covariance matrices of each distribution.

The first step of the algorithm is the expectation step or "E" step. Here the expected value of the complete-data log likelihood is update by replacing the z_{ig} in l_c by their expected values conditioned on the value of the parameter estimates at the current iteration of the algorithm. These expected values are given by

$$\mathbb{E}[Z_{ig}|\mathbf{x}_i] = \frac{\pi_g \phi(\mathbf{x}_i|\mu_g, \mathbf{\Sigma}_g)}{\sum_{h=1}^{G} \pi_h f(\mathbf{x}_i|\mu_g, \mathbf{\Sigma}_g)} =: \hat{z}_{ig}$$

for $i = 1, ..., n$ and $g = 1, .., G$. (S. McNicholas 2021)

The second step is the maximization step or "M" step. Here the model parameters are updated by maximizing the expected value of the complete data log likelihood after updating it with the new expected values of the indicators z_{ig}. This maximization happens with respect to π_g, $\mu_\mathbf{g}$, and $\mathbf{\Sigma_g}$ which gives the updates

$$\hat{\pi}_g = \frac{n_g}{n}, \hat{\mu}_{\mathbf{g}} = \frac{\sum_{i=1}^{n} \hat{z}_{ig}\mathbf{x}_i}{n_g}, \hat{\mathbf{\Sigma}}_g = \frac{1}{n_g}\sum_{i=1}^{n} \hat{z}_{ig}(\mathbf{x}_i - \hat{\mu}_g)(\mathbf{x}_i - \hat{\mu}_g)'$$

These two steps alternate until the stopping criterion is met.

One popular choice of stopping criterion based on the Aitken's acceleration [aitken,1926] which is defined as

$$a^{(k)} = \frac{l^{(k+1)} - l^{(k)}}{l^{(k)} - l^{(k-1)}}$$

Where k is the current iteration number. It is used to calculate an asymptotic estimate of the log-likelihood starting at the next iteration which is a method of checking to see whether or not the algorithm is in a maximum for which it is likely to escape if it continues to run. This asymptotic estimate is given by

$$l_\infty^{(k+1)} = l^{(k)} + \frac{l^{(k+1)} - l^{(k)}}{1 - a^{(k)}}$$

A choice of convergence criteria includes when the following hold

$$\mid l_\infty^{(k+1)} - l_\infty^{(k)} \mid < \epsilon$$

$$l_\infty^{(k)} - l^{(k)} < \epsilon$$

$$l_\infty^{(k+1)} - l^{(k)} < \epsilon$$

(S. McNicholas 2021)

In using the EM algorithm to fit a GPCM to the data clusters, one must specify the number of normal distributions to use in the fit, which is a tuneable parameter which is used tuned by the Bayesian Information Criterion (BIC)

$$BIC = 2l(\mathbf{x}, \hat{\vartheta}) - \rho \log(n)$$

Where $\hat{\vartheta}$ is the MLE of ϑ, ρ is the number of free parameters, and n is the number of observations.

For my usage of the EM algorithm on a GPCM to cluster the glass data, I started by scaling the glass data. I then ran the GPCM for 6 clusters in the mixture as this is as many types of glass there are in the dataset. During this, I chose a k-means start to initialize the positions of the clusters at the beginning of the EM algorithm because I found a marginal improvement in the final classification doing this versus doing a random start. This GPCM was done using the R mixture library. (P. D. McNicholas 2021-03-02)
The selected model by using the BIC has 6 clustering components and uses the VEE covariance structure. This is the summary of the BIC values for the tested covariance structures.

	x
VEE	-3049.221
VVE	NA
EVE	NA
EEE	-3920.273
VVI	NA
EVI	NA
VEI	-3824.976

	x
EEI	-5154.077
VII	-4439.959
EII	-5096.759

The following is the resulting confusion matrix of the clustering.

	1	2	3	4	5	6
1	28	0	25	14	2	1
2	7	10	35	15	8	1
3	9	0	4	2	1	1
5	0	7	0	0	3	3
6	0	3	0	0	5	1
7	0	21	0	0	3	5

The misclassification rate of the clustering is 0.7570093 and the adjusted rand is 0.1701606.

Results

It is clear from the random forest results that it is possible to classify the type of an unknown piece of glass by measuring its chemical composition and its refractive index. The results of the random forest of a misclassification rate of 0.02803738 and adjusted rand index 0.9395585 seem to be effective enough to allow good identification of glass so as it may be used as evidence in court. The GPCM trained via the EM algorithm however did not do nearly as well with misclassification rate of 0.7570093 and adjusted rand is 0.1701606. This model is unlikely to help identify glass for use in court.
It is possible that the GPCM was hindered by divergences from normality in the distributions of the variables in addition to the close overlap of some of the predictor variables as we saw in the descriptive analysis.

Conclusions

The promising results of the random forest in classifying the glass indicates that classifying unknown pieces of glass based on these measures of physical characteristics and chemical composition could be best accomplished by a nonparametric classifier like random forest, perhaps further exploration of clustering using k-nearest neighbours, k-means, or perhaps hierarchical clustering could provide a more robust and extendable framework for glass identification on a larger scale.

References

German, B. 1987. “Glass Identification Data Set.” USA Forensic Science Service. https://archive.ics.uci.edu/ml/datasets/Glass+Identification.

Leo Breiman, Fortran original by, R port by Andy Liaw Adele Cutler, and Matthew Wiener. 2018-03-22. “Breiman and Cutler’s Random Forests for Classification and Regression.” 4.6-14. `https://www.stat.berkeley.edu/~breiman/RandomForests/`.

McNicholas, Paul D. 2021-03-02. “Mixture Models for Clustering and Classification.” 2.0.3. mcnicholas@math.mcmaster.ca.

McNicholas, Sharon. 2021. “Class Notes.” McMaster University.