Data Analysis and Classification Modelling on Chemical Data

2023-12-19

Missing values

The first step was to examine the proportion of data which was missing. Table 1 shows which variables contained missing data and the number of observations which contained a missing value in that variable.

variable	n_miss	pct_miss
V10	6	0.24
V9	5	0.20
V20	5	0.20
V2	3	0.12
V3	3	0.12
V7	3	0.12
V8	3	0.12
V12	3	0.12
V18	3	0.12
V1	2	0.08
V6	2	0.08
V11	2	0.08
V15	2	0.08
V16	2	0.08
V19	2	0.08
V4	1	0.04
V13	1	0.04
V14	1	0.04
V17	1	0.04

Table 1 shows the variables which contained missing data and the proportions of missing data in those variables.

As only 2 percent of the data were missing, only the observations which had complete cases were used for the dimension reduction and the classification models.

The data was first split into training, test and validation sets with a split of 50% training, 25% test and 25% validation. Functions from the base R (2022a) were used for this. Each observations was assigned a letter with 50% getting A, representing the training set. 25% of the dataset were assigned B, and 25% were assigned C, representing the test and validation sets. This ensured that each observation was only put into one of the sets.

Exploratory Data Analysis on the training set

The data shows a relatively even distribution between the classes (A-E), therefore no resampling was needed for this data.

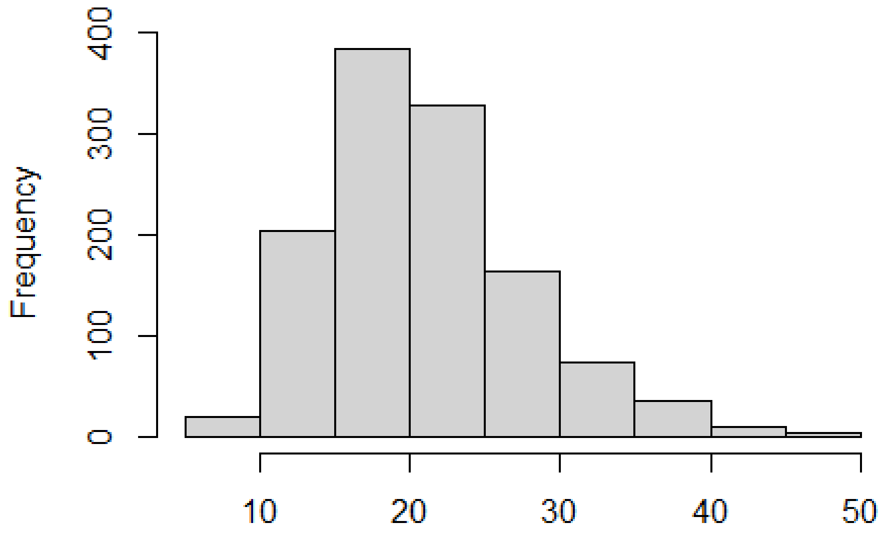
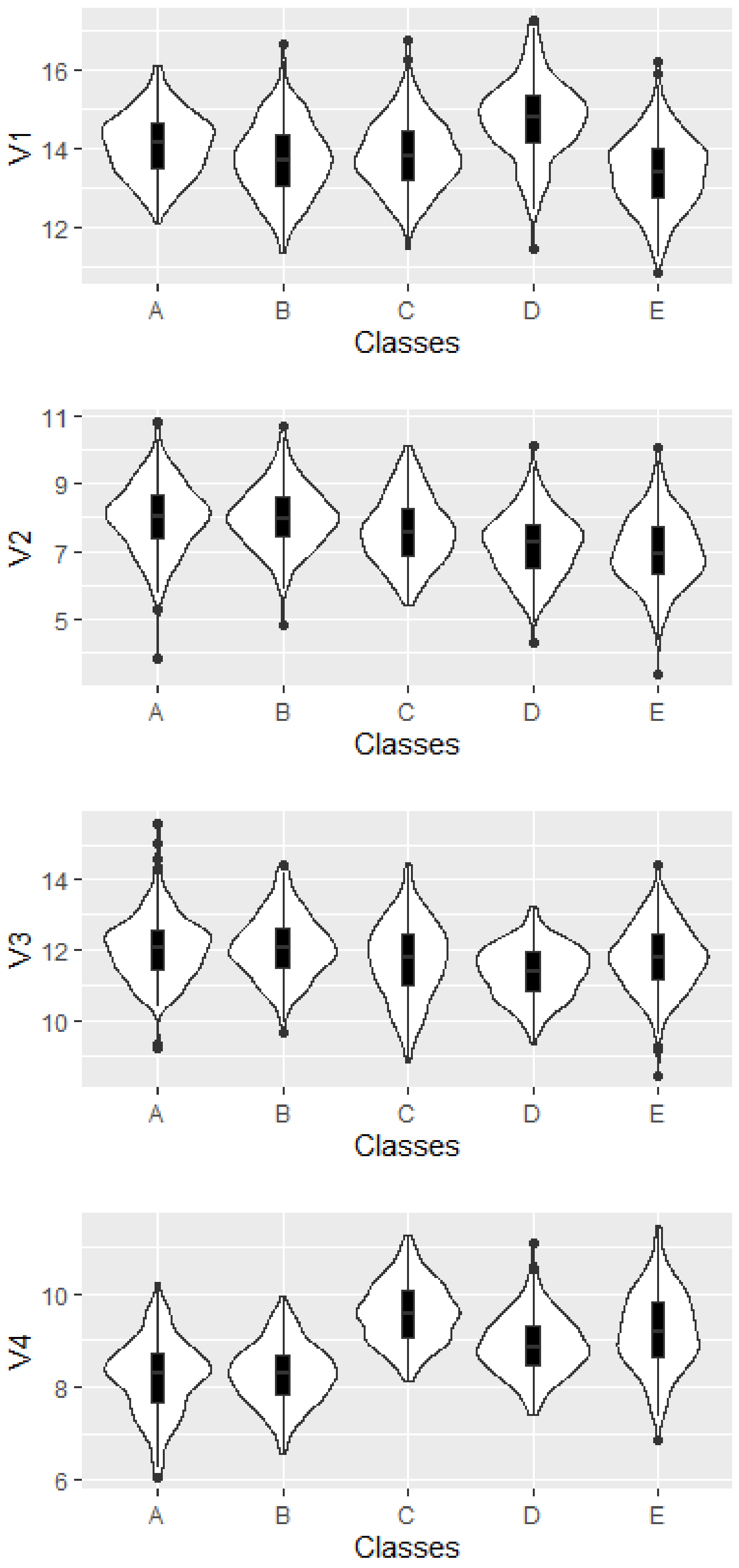
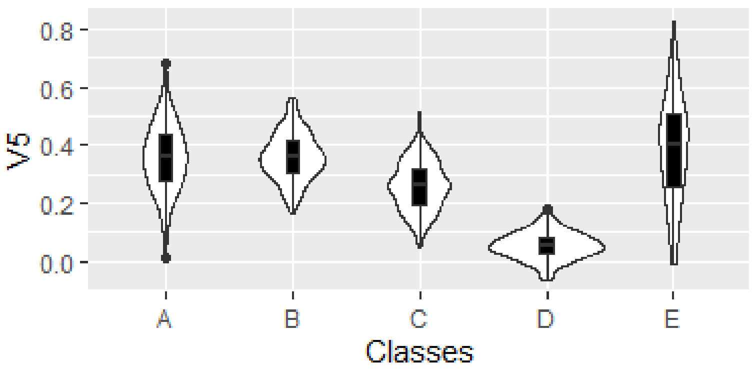
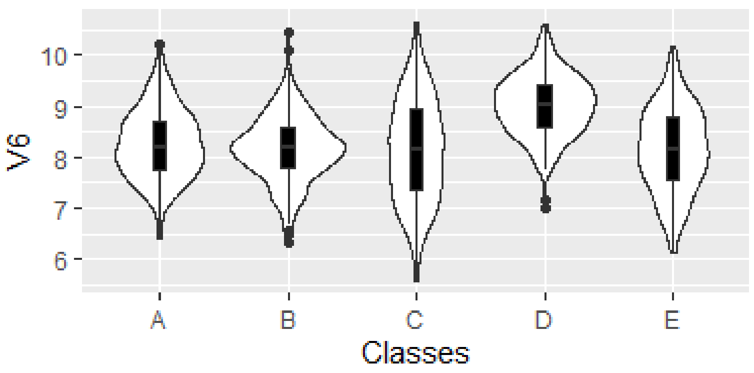


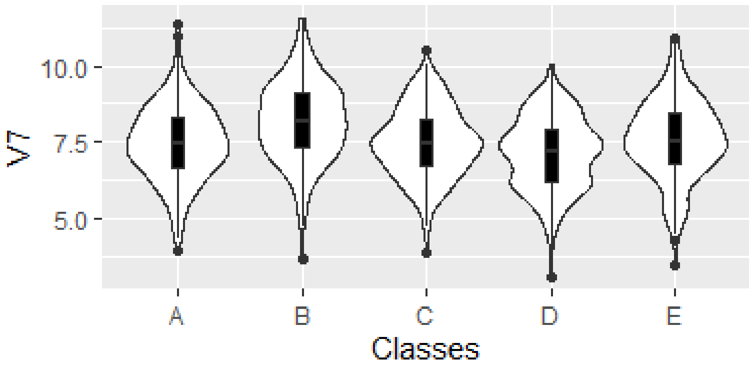
Figure 1. The average mahalanobis distance for each row in the training set. The `maha_dist` function from the assertr package (Fischetti, 2022) was used to measure this.

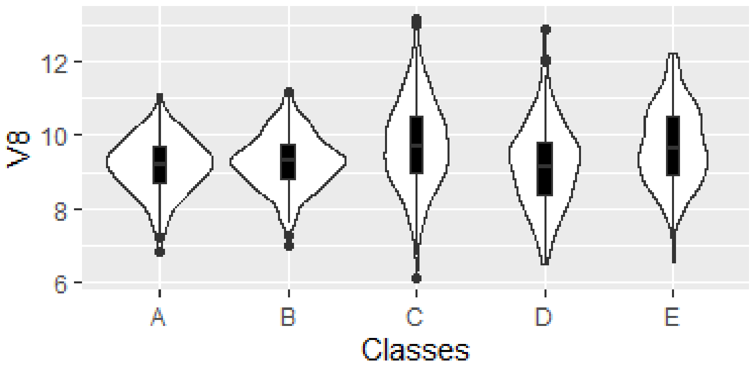
The larger values along the x axis could be potential outliers. However, more information on the features of the data would be needed to determine the expected values for each variable and decide whether these would be considered outliers. Figure 1 shows that there are no extreme outliers in the training set.

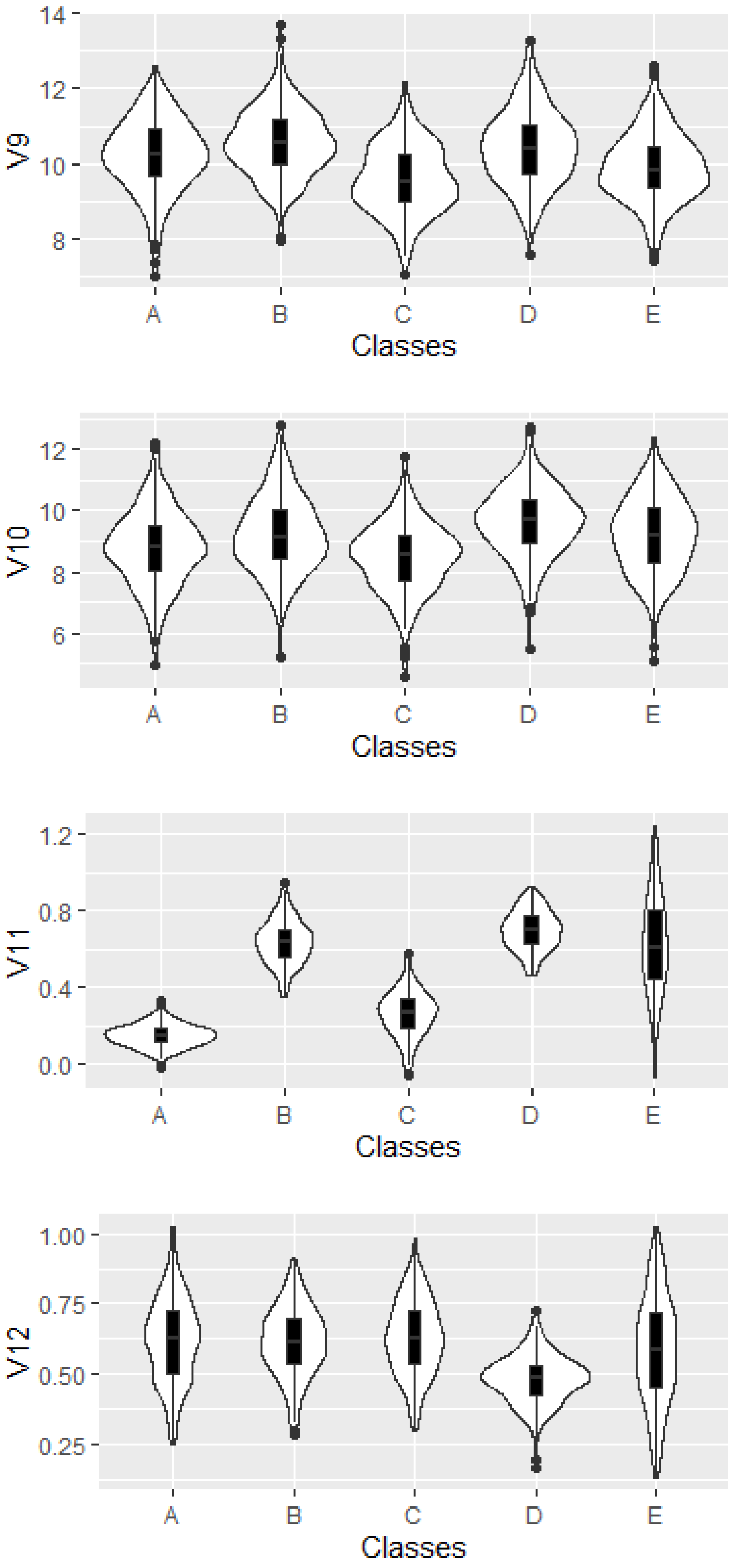

V1
16
14
12
A
B
C
D
E
Classes
V2
11
9
7
5
A
B
C
D
E
Classes
V3
14
12
10
A
B
C
D
E
Classes
V4
10
8
6
A
B
C
D
E
Classes

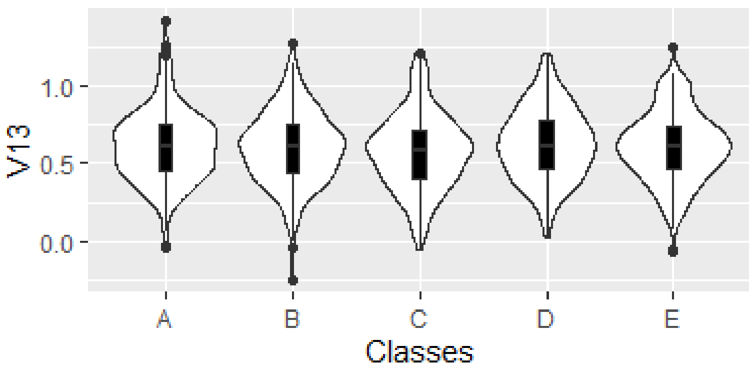

V5
0.8
0.6
0.4
0.2
0.0
A
B
C
D
E
Classes

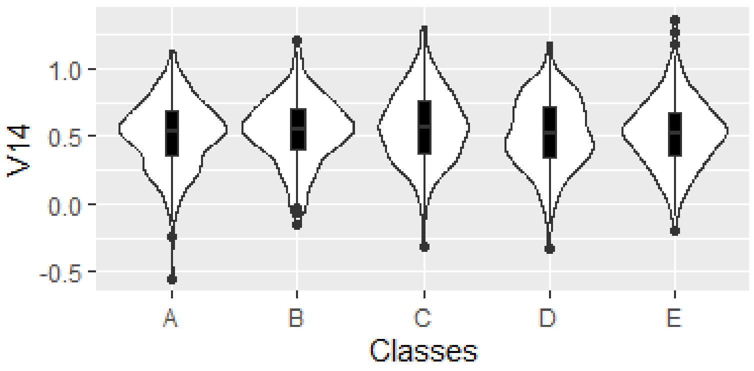

V6
10
9
8
7
6
A
B
C
D
E
Classes

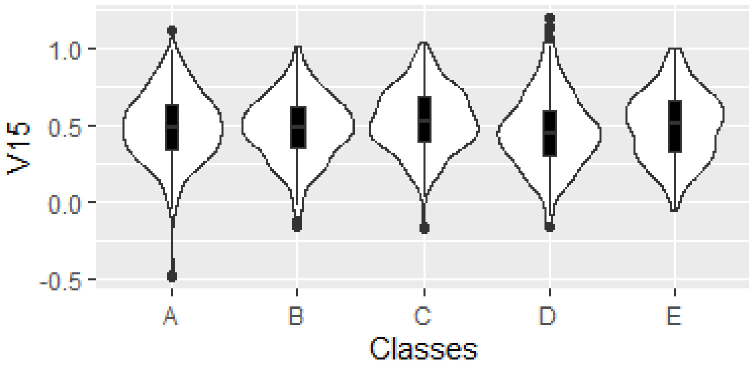

V7
10.0
7.5
5.0
A
B
C
D
E
Classes

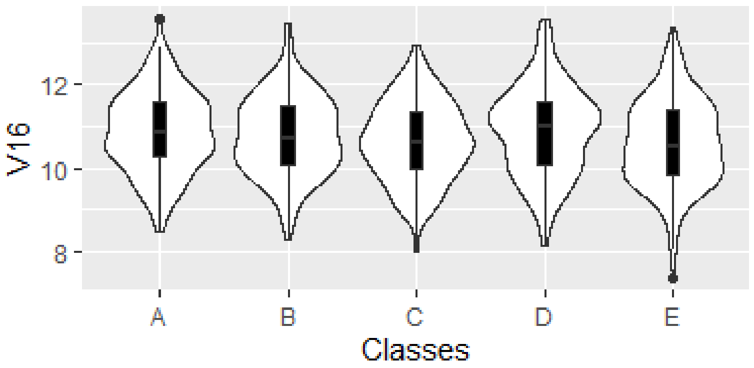

V8
12
10
8
6
A
B
C
D
E
Classes


V9
14
12
10
8
A
B
C
D
E
Classes
V10
12
10
8
6
A
B
C
D
E
Classes
V11
1.2
0.8
0.4
0.0
A
B
C
D
E
Classes
V12
1.00
0.75
0.50
0.25
A
B
C
D
E
Classes


V13
1.0
0.5
0.0
A
B
C
D
E
Classes


V14
1.0
0.5
0.0
-0.5
A
B
C
D
E
Classes


V15
1.0
0.5
0.0
-0.5
A
B
C
D
E
Classes


V16
12
10
8
A
B
C
D
E
Classes

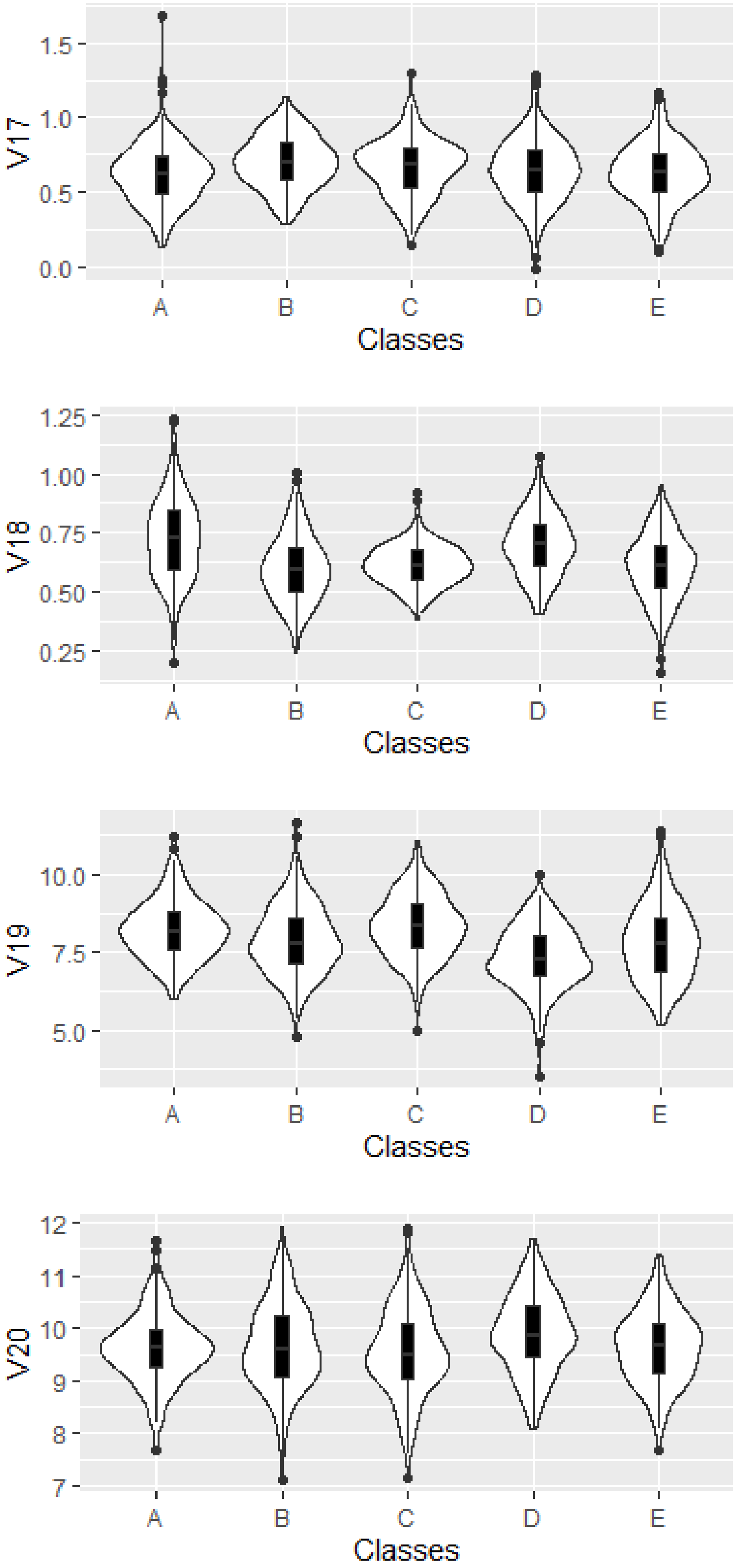


Figure 2. Violin plots and box plots which show the distribution of each variable and each class.

From Figure 2, the majority of the variables appear to be normally distributed with few outliers. There are some negative values in variables V5, V11, V13, V14, V15 and V17. Many of the negative values show to be outliers for their class with the rest of the values being positive. This should be noted. This may be common for this type of data, more domain knowledge would be needed to identify if these are concerning.

From Figure 1, we can see that V5 could be an indicator that an observation belongs to class D as they appear to cluster with a lower mean and smaller spread than the other classes. V11 appears to have a large difference in the mean and spread of each class. It appears to have a bi modal distribution overall. V11 could be an indicator that an observation belongs to class A, as these seem to cluster with a lower mean and smaller spread than the other classes.

The plots in Figure 2 have been created using the ggplot2 (Wickham, 2016) package.

variable	Class	n	mean	median	sd	min	max
V1	A	236	14.09000	14.13000	0.79760	12.09000	16.1400
	B	272	13.72000	13.72000	0.94590	11.36000	16.6600
	C	234	13.84000	13.80000	0.88770	11.43000	16.7700
	D	232	14.75000	14.79000	0.99020	11.45000	17.2600
	E	251	13.40000	13.42000	0.92590	10.86000	16.2200
V2	A	236	7.97900	8.02100	1.01400	3.84400	10.8200
	B	272	7.99900	7.97100	0.91330	4.82000	10.7000
	C	234	7.60400	7.53600	1.00200	5.38600	10.1100
	D	232	7.18300	7.24200	0.96040	4.29700	10.1600
	E	251	7.01100	6.94100	1.03900	3.38100	10.0900
V3	A	236	12.07000	12.06000	0.91360	9.22600	15.5800
	B	272	12.11000	12.06000	0.88920	9.69800	14.4300
	C	234	11.73000	11.81000	1.10700	8.84400	14.4900
	D	232	11.40000	11.43000	0.74640	9.36200	13.2700
	E	251	11.78000	11.79000	0.93850	8.47900	14.4200
V4	A	236	8.22100	8.30200	0.78330	6.07300	10.2400
	B	272	8.26900	8.29000	0.65130	6.57500	9.9540
	C	234	9.58700	9.57400	0.65010	8.12200	11.2400
	D	232	8.89800	8.85500	0.62810	7.39200	11.0700
	E	251	9.21300	9.18500	0.85170	6.87200	11.4600
V5	A	236	0.35800	0.35980	0.11840	0.01180	0.6806

variable	Class	n	mean	median	sd	min	max
	B	272	0.36120	0.35780	0.08247	0.16270	0.5664
	C	234	0.25780	0.26110	0.08349	0.04490	0.5153
	D	232	0.05383	0.05235	0.04586	-0.06201	0.1793
	E	251	0.38790	0.40030	0.17870	-0.01137	0.8272
V6	A	236	8.26400	8.19800	0.68390	6.40100	10.2200
	B	272	8.16600	8.18300	0.65150	6.33200	10.4400
	C	234	8.12100	8.14900	1.00700	5.58900	10.6400
	D	232	9.01300	9.03900	0.59670	6.98700	10.6100
	E	251	8.13700	8.15900	0.83130	6.11600	10.1800
V7	A	236	7.47700	7.44400	1.24900	3.94300	11.3800
	B	272	8.17900	8.16800	1.30400	3.64900	11.5500
	C	234	7.47000	7.44600	1.15000	3.86100	10.5100
	D	232	7.12700	7.19300	1.18100	3.08900	10.0100
	E	251	7.57400	7.50100	1.30700	3.43100	10.9300
V8	A	236	9.17600	9.20600	0.74420	6.83500	11.1000
	B	272	9.27500	9.28400	0.74470	6.97800	11.1900
	C	234	9.73500	9.69100	1.18400	6.14800	13.1400
	D	232	9.17500	9.16900	1.10900	6.52000	12.8600
	E	251	9.70000	9.61600	1.06600	6.57500	12.2100
V9	A	236	10.22000	10.26000	0.97390	7.01400	12.5900
	B	272	10.58000	10.55000	0.90060	7.95800	13.6900
	C	234	9.58500	9.52200	0.90410	7.02200	12.1700
	D	232	10.42000	10.41000	0.98990	7.55700	13.2800
	E	251	9.89500	9.84700	0.93770	7.39600	12.6000
V10	A	236	8.77000	8.79700	1.23000	4.96800	12.1900
	B	272	9.22500	9.15900	1.23300	5.17800	12.7900
	C	234	8.48200	8.55200	1.12200	4.58200	11.7600
	D	232	9.62000	9.71000	1.12200	5.48700	12.7300
	E	251	9.18500	9.21600	1.27100	5.06600	12.4200
V11	A	236	0.14850	0.14940	0.05991	-0.02063	0.3305

variable	Class	n	mean	median	sd	min	max
	B	272	0.63510	0.63320	0.11710	0.34630	0.9455
	C	234	0.26330	0.27360	0.12040	-0.05954	0.5814
	D	232	0.69720	0.69520	0.10190	0.46050	0.9239
	E	251	0.62170	0.60590	0.25710	-0.06558	1.2410
V12	A	236	0.61950	0.62760	0.15080	0.24850	1.0240
	B	272	0.61350	0.61210	0.12400	0.28170	0.9126
	C	234	0.62890	0.62870	0.13500	0.29580	0.9807
	D	232	0.48060	0.48550	0.08798	0.16590	0.7248
	E	251	0.58590	0.58480	0.19080	0.13390	1.0240
V13	A	236	0.61360	0.60930	0.23130	-0.03327	1.4140
	B	272	0.60170	0.61260	0.23120	-0.24190	1.2740
	C	234	0.57550	0.58410	0.24520	-0.05515	1.2050
	D	232	0.62750	0.61610	0.23080	0.02198	1.2130
	E	251	0.60910	0.61060	0.22920	-0.06109	1.2530
V14	A	236	0.50790	0.53520	0.25190	-0.54790	1.1250
	B	272	0.53560	0.55440	0.24010	-0.14500	1.2070
	C	234	0.56810	0.56740	0.26770	-0.32010	1.3080
	D	232	0.51840	0.51550	0.25940	-0.32480	1.1910
	E	251	0.52040	0.52420	0.25020	-0.19140	1.3530
V15	A	236	0.48410	0.48570	0.23460	-0.48520	1.1210
	B	272	0.47700	0.48310	0.21070	-0.16200	1.0130
	C	234	0.52440	0.52570	0.21050	-0.17750	1.0390
	D	232	0.46050	0.44330	0.24280	-0.16530	1.1920
	E	251	0.49140	0.50630	0.21930	-0.05399	0.9939
V16	A	236	10.86000	10.85000	0.95600	8.47000	13.5400
	B	272	10.78000	10.72000	0.95540	8.30500	13.4500
	C	234	10.64000	10.61000	0.94110	7.98900	12.9200
	D	232	10.89000	11.00000	1.05300	8.13500	13.5500
	E	251	10.61000	10.54000	1.03200	7.40100	13.3500
V17	A	236	0.62360	0.62200	0.20640	0.13200	1.6830

variable	Class	n	mean	median	sd	min	max
	B	272	0.70430	0.70370	0.17620	0.28630	1.1460
	C	234	0.66640	0.68500	0.18940	0.13680	1.3060
	D	232	0.64390	0.64300	0.21440	-0.01253	1.2850
	E	251	0.62840	0.62740	0.19930	0.10270	1.1630
V18	A	236	0.71970	0.72430	0.17050	0.19310	1.2300
	B	272	0.59540	0.59190	0.13990	0.23840	1.0020
	C	234	0.61110	0.60800	0.08566	0.38580	0.9221
	D	232	0.70030	0.69740	0.13260	0.40470	1.0710
	E	251	0.60220	0.61310	0.14190	0.15790	0.9513
V19	A	236	8.21600	8.19000	0.93930	6.02900	11.2200
	B	272	7.89400	7.79100	1.09000	4.78600	11.6400
	C	234	8.33700	8.33300	1.01500	4.98500	11.0900
	D	232	7.33000	7.28800	0.99410	3.55400	9.9850
	E	251	7.80100	7.77400	1.21500	5.19700	11.3800
V20	A	236	9.60900	9.62000	0.64450	7.67700	11.6600
	B	272	9.65400	9.58700	0.81270	7.12000	11.9300
	C	234	9.55300	9.50000	0.82100	7.14400	11.9100
	D	232	9.89000	9.87200	0.74160	8.08400	11.7200
	E	251	9.63200	9.65900	0.68430	7.65900	11.3900

Table 2. The number of observations, mean, median, standard deviation, min and max for each variable separated by each class.

Table 2 Allows the structure of the data to be viewed, the scale for each variable and any differences between the descriptive statistics between the groups but within variables. Again, we can notice the negative values in the data, these may be of concern, but more information about the data would be needed to determine if negative values are appropriate for these variables.

This has been created using the `describeBy` function from the (Revelle, 2022) package to produce a matrix of descriptive statistics from the data grouped by the label (A,B,C,D,E) of the observation. From the base R (2022a) package `cbind` is used to combine the names of the variables for the matrix of descriptive statistics for each label. Flextable (Gohel and Skintzos, 2023) is used to create the table with formatting.

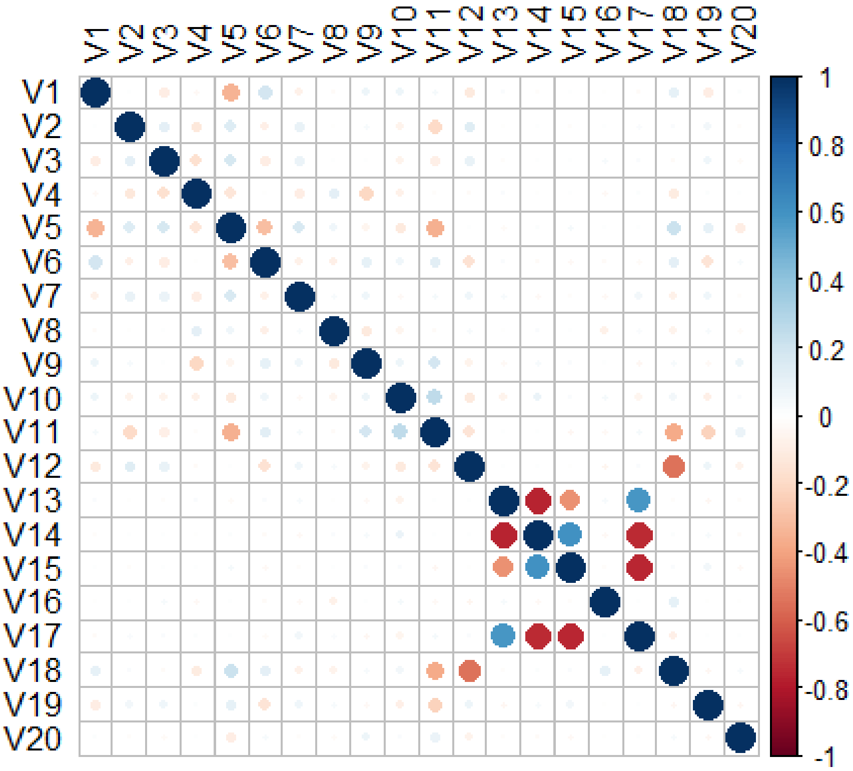


Figure 3. The correlation plot of the variables in the training set. Produced using the stats (2022b) and the corrplot (Wei and Simko, 2021) packages.

From the correlation plot in Figure 3, we can see that there are a few variables which are correlated. Variable V14 with V13 and V17 appear to be strongly negatively correlated. Variable V17 and V15 also appear to be strongly negatively correlated. Variables V14 and V15 appear to have moderate positive correlation with each other. This means that there are possibles relationships between the variables and means we can potentially reduces the dimensionality of the dataset using Principal Component Analysis (PCA).

Principal Components Analysis

Performing parallel analysis prior to PCA shows that 5 dimensions should be retained, as these had eigenvalues greater than one. This shows how much variation is being explained more than by chance, and so how many dimensions to retain to explain this variation.

The `dudi.pca` function from the (Dray and Dufour, 2007) package was used to perform the PCA, and functions from the (Kassambara and Mundt, 2020) package were used to get the eigen values and plot results of the PCA.

The true labels are returned to the observations using the `cbind` function from base R (2022a).

Variables	CS1	CS2	CS3	CS4	CS5
V1	-0.0006020	-0.3093490	-0.1053531	-0.0174398	0.5038405

Variables	CS1	CS2	CS3	CS4	CS5
	897	83	4	60	2
V2	-0.0021956202	0.237357933	-0.07587535	0.238750686	0.39581254
V3	0.0030709037	0.243305488	-0.02262969	0.301808796	0.03707147
V4	-0.0180229303	-0.052759025	0.18219342	-0.562099027	-0.08205198
V5	0.0294636039	0.460207861	-0.24514396	0.070035247	-0.30439980
V6	-0.0099300368	-0.357689016	-0.11273034	0.003327951	0.27098245
V7	-0.0319980080	0.162741151	0.06814091	0.322632424	-0.19624293
V8	-0.0033351139	0.112158249	0.13872116	-0.245705459	-0.16193494
V9	0.0368365368	-0.169703672	-0.08173107	0.482804051	0.03861570
V10	0.0558652981	-0.257306047	-0.03145258	0.180311180	-0.31880347
V11	-0.0292761832	-0.398610306	0.31725565	0.241562635	-0.34261423
V12	-0.0208097555	0.254077266	0.48694560	0.113828281	0.32006355
V13	-0.4684274803	-0.004843731	-0.05114995	-0.024285180	0.02947491
V14	0.5259356687	-0.002456982	0.06417586	0.004144140	-0.01490742
V15	0.4700091409	0.020774574	0.02487506	-0.028191715	0.02978961
V16	-0.0177567	-0.0082412	-0.1714338	0.053500023	-0.0450601

Variables	CS1	CS2	CS3	CS4	CS5
	156	61	2		3
V17	-0.5207024048	0.034584478	0.00704370	0.023415706	-0.04131113
V18	0.0493635025	-0.021095396	-0.68489416	-0.117323899	-0.03781320
V19	0.0307290824	0.260005860	0.02517102	-0.063453142	0.12807230
V20	-0.0110589666	-0.137702936	-0.02835284	0.082137140	-0.06360498

Table 3 The linear combination of the variables for each principal component created by the PCA.

All of the original variables are represented by the principal components. Each value in Table 3 represents how the that variable is contributing to the result of the observation (in this case, the class label the observation is given).

Principal component 1 is responsible for capturing the largest variation in the data. It shows a linear combination of the variables. The variables 13, 14, 15 and 17 are captured the most by principal component 1. The high negative correlation between variables 14 and 17 is captured here by showing that they have similar magnitudes in the opposite directions for their column normed scores. The high negative correlation between 17 and 15, and 13 and 14 are also captured by principal component 1, as well as the positive correlation between variables 14 and 15. These correlations were shown to be present in the original training dataset, see Figure 3. This means that principal component 1 can represent the results covered by these variables and reduces the need for all four raw variables to be in the data.

Principal component 2 captures variable 5 in one direction and variable 6 in the other direction. Principal component 3 predominantly describes variable 18, and variable 12 to a slightly lesser extent in the opposite direction. Principal component 4 is capturing variable 4. It is also capturing variables 9, 3 and 7 to a lesser extent in the opposite direction. Principal component 5 is predominantly capturing variable 1. It is also capturing variable 2 and 12 in the same direction and 11 and 5 in the opposite direction.

Here we have the results of all 20 variables captured by 5 dimensions. This shows that we can still capture the same information from the observations but with fewer dimensions.

To perform the same PCA that was performed on the training set on the test and validation sets: the test and validation was first mean centered and scaled. Matrix multiplication was performed between these scaled datasets and the column normed scores from the PCA of the training set. This gives the values of the training and validation sets projected onto the PCA. This was then converted to a dataframe and the labels combined to the data frame.

Classification models

Five different classification models were trained and tested. The performance of the models was compared using the accuracy metric. Firstly, a Random Forest model was developed using the randomForest (Liaw and Wiener, 2002) package. Hyperparameter tuning was performed to asses the number of trees which should be used in the random forest model and the optimal number of variables to be considered at each split. To evaluate model performance the `confusionMatrix` function from the caret package (Kuhn, 2023) was used to create confusion matrices for each model. The class package (Venables and Ripley, 2002a) was used to create the K Nearest Neighbors model (KNN). The KNN model was tested using from one to fifty neighbors (k) to obtain the k value which produced the optimal model performance. The Mclust package (Scrucca *et al.*, 2016) was used to perform model based discriminant analysis (MBDA). The MASS (Venables and Ripley, 2002b) package was used for linear discriminant analysis (LDA) and quadratic discriminant analysis (QDA).

Accuracy was chosen as it gives an overall representation of the performance of the model. The accuracy is calculated by the total number of correct predictions over the total number of observations.

Model	Accuracy
Random Forest	86.93
KNN	70.92
MBDA	72.39
LDA	73.20
QDA	73.04

Table 4. The in-sample performance for each of the classification models tested. The accuracy is measured as a percentage. KNN: k nearest neighbors. MBDA: model based discriminant analysis. LDA: linear discriminant analysis. QDA: quadratic discriminant analysis.

Table 4 shows the best performing model to be the Random Forest model with an accuracy of 86.93% on the in-sample test data. This model had a much higher accuracy than the other models. The model also had a greater performance on the out-of-sample validation set, with an accuracy of 89.89%. This suggests that the Random Forest model is not overfitting the training data and demonstrates how it performs on unseen data. The Random Forest model was performed on all features of the data, without prior PCA. Therefore, for optimal classification performance, the findings from this report suggests that all variables that were present in the original dataset should be measured.

Dray, S. and Dufour, A. (2007) The ade4 package: Implementing the duality diagram for ecologists. [online]. 22.

Fischetti, T. (2022) Assertr: Assertive programming for r analysis pipelines. [online]. Available from: https://CRAN.R-project.org/package=assertr.

Gohel, D. and Skintzos, P. (2023) *Flextable: Functions for tabular reporting* [online]. Available from: https://CRAN.R-project.org/package=flextable.

Kassambara, A. and Mundt, F. (2020) Factoextra: Extract and visualize the results of multivariate data analyses. [online]. Available from: https://CRAN.R-project.org/package=factoextra.

Kuhn, M. (2023) *Caret: Classification and regression training* [online]. Available from: https://github.com/topepo/caret/.

Liaw, A. and Wiener, M. (2002) Classification and regression by randomForest. [online]. 2, pp.18–22. Available from: https://CRAN.R-project.org/doc/Rnews/.

R Core Team (2022a) R: A language and environment for statistical computing. [online]. Available from: https://www.R-project.org/.

R Core Team (2022b) R: A language and environment for statistical computing. [online]. Available from: https://www.R-project.org/.

Revelle, W. (2022) Psych: Procedures for psychological, psychometric, and personality research. [online]. Available from: https://CRAN.R-project.org/package=psych.

Scrucca, L., Fop, M., Murphy, T.B. and Raftery, A.E. (2016) Mclust 5: Clustering, classification and density estimation using gaussian finite mixture models. [online]. 8. Available from: https://doi.org/10.32614/RJ-2016-021.

Venables, W.N. and Ripley, B.D. (2002a) Modern applied statistics with s. [online]. Available from: https://www.stats.ox.ac.uk/pub/MASS4/.

Venables, W.N. and Ripley, B.D. (2002b) Modern applied statistics with s. [online]. Available from: https://www.stats.ox.ac.uk/pub/MASS4/.

Wei, T. and Simko, V. (2021) R package 'corrplot': Visualization of a correlation matrix. [online]. Available from: https://github.com/taiyun/corrplot.

Wickham, H. (2016) ggplot2: Elegant graphics for data analysis. [online]. Available from: https://ggplot2.tidyverse.org.