# Advanced Statistics: Application of supervised and unsupervised methods to biological data

Daniel Hill

2025-04-27

**Abstract:** Biological data with twenty features and four categorical class labels is explored and analysed using advanced statistical techniques in this report. Both supervised and unsupervised methods were implemented and evaluated, and the broad selection of models includes logistic regression, support vector machine, random forest, agglomerative hierarchical clustering, and gaussian mixture modelling. Comparing the results achieved using a selection of models with different underlying principles gives insight into the nature of the data. For example, the success of model-based clustering compared to hierarchical clustering and tree-based learning suggests the lack of hierarchy among the categorical labels, and the success of factor analysis as a dimensionality reduction technique suggests the presence of underlying biological mechanisms leading to several of the features arising together. Models achieving over 90% accuracy were produced, but all models performed notably worse at separating one of the categories that overlapped the other three.

# Contents



# List of Figures

## List of Tables

# 1 Introduction

This report takes a moderately sized biological dataset containing 3000 observations, where each observation having 20 numeric varables and one catagorical label. These data are explored thoroughly before being used to train and test an array of statistical modelling techniques.

This is an interesting project because the origin and meaning of the variables in the data are completely unknown - an unusual scenario in the data science field, where usually it is the domain knowledge and problem context that inform the selection and implementation of statistical methods. Here, with this relationship reversed, algorithms have been chosen so that the evaluation of their performance can attempt to uncoverthe underlying biological significance of the variables.

Achieving meaningful results in this task shows the importance of supervised and unsupervised learning to this field, where classification algorithms can build valuable models that have high impact on society such as disease diagnosis models, and unsupervised learning techniques can create breakthoughs in identifying clusters of data that lead to new discoveries and classifications.

# 2 Methods

## 2.1 Data Description

Each of the 3000 observations has 20 numeric features and a label placing it in one of four catagorical classes. Though exploritory data analysis, two groups of correlated features were identified. Outliers were identified and removed using z score method. One feature transformed using logarithm to create a more normal distribution to improve the performance of models. All features were scaled and centered. After the preprocessing of the data, 2776 usable observations remained.

Bootstrap sampling was used to create a larger dataset so that the performance of the models could be compared between the original and bootstrapped data.

## 2.2 Exploratory Data Analysis Approach

find distributions within each feature, look for correlations between features, scatter between plots that features that have high correlation to the catagorical label or another feature, use PCA to visualize all data together, calculate hopkins statistic to determine the clustering tendency of the data

Table 1: Feature Descriptions

| Variable Name | No. missing values | mean | Std deviation | min | 25th %ile | median | 75th %ile | max |
|---|---|---|---|---|---|---|---|---|
| X1 | 2 | 9.876 | 0.764 | 6.840 | 9.356 | 9.872 | 10.399 | 12.355 |
| X2 | 0 | 10.151 | 1.040 | 6.538 | 9.445 | 10.138 | 10.855 | 14.021 |
| X3 | 1 | 8.861 | 0.871 | 6.424 | 8.243 | 8.847 | 9.445 | 12.216 |
| X4 | 2 | 8.939 | 1.275 | 3.875 | 8.088 | 8.926 | 9.805 | 13.351 |
| X5 | 0 | 13.853 | 0.942 | 10.527 | 13.236 | 13.858 | 14.492 | 16.557 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| X6 | 0 | 8.151 | 1.026 | 4.815 | 7.447 | 8.134 | 8.856 | 11.871 |
| X7 | 1 | 0.426 | 0.278 | 0.000 | 0.185 | 0.375 | 0.678 | 1.301 |
| X8 | 3 | 0.234 | 0.197 | 0.000 | 0.102 | 0.170 | 0.300 | 1.230 |
| X9 | 0 | 0.717 | 0.247 | 0.006 | 0.532 | 0.751 | 0.888 | 1.679 |
| X10 | 0 | 0.378 | 0.155 | 0.000 | 0.281 | 0.372 | 0.469 | 1.199 |
| X11 | 1 | 9.175 | 1.087 | 6.031 | 8.412 | 9.077 | 9.860 | 13.027 |
| X12 | 0 | 11.930 | 0.977 | 8.046 | 11.290 | 11.913 | 12.594 | 15.478 |
| X13 | 1 | 8.228 | 0.806 | 4.919 | 7.719 | 8.244 | 8.746 | 11.226 |
| X14 | 2 | 7.846 | 1.238 | 3.574 | 7.022 | 7.884 | 8.689 | 12.413 |
| X15 | 1 | 10.701 | 0.962 | 7.572 | 10.054 | 10.701 | 11.344 | 14.037 |
| X16 | 1 | 7.814 | 1.052 | 3.801 | 7.132 | 7.830 | 8.521 | 11.668 |
| X17 | 2 | 0.504 | 0.221 | 0.001 | 0.348 | 0.502 | 0.653 | 1.315 |
| X18 | 3 | 0.682 | 0.204 | 0.004 | 0.544 | 0.686 | 0.819 | 1.390 |
| X19 | 0 | 0.544 | 0.254 | 0.000 | 0.363 | 0.545 | 0.710 | 1.518 |
| X20 | 0 | 0.589 | 0.231 | 0.012 | 0.434 | 0.587 | 0.746 | 1.353 |

This description of the predictive features shows the range of scales, ranging by and order of magnitude. Several models such as support vector machine analysis are affected by the scale and centering of the data it learns from, so it this was identified as an important preprocessing step that had to be performed.

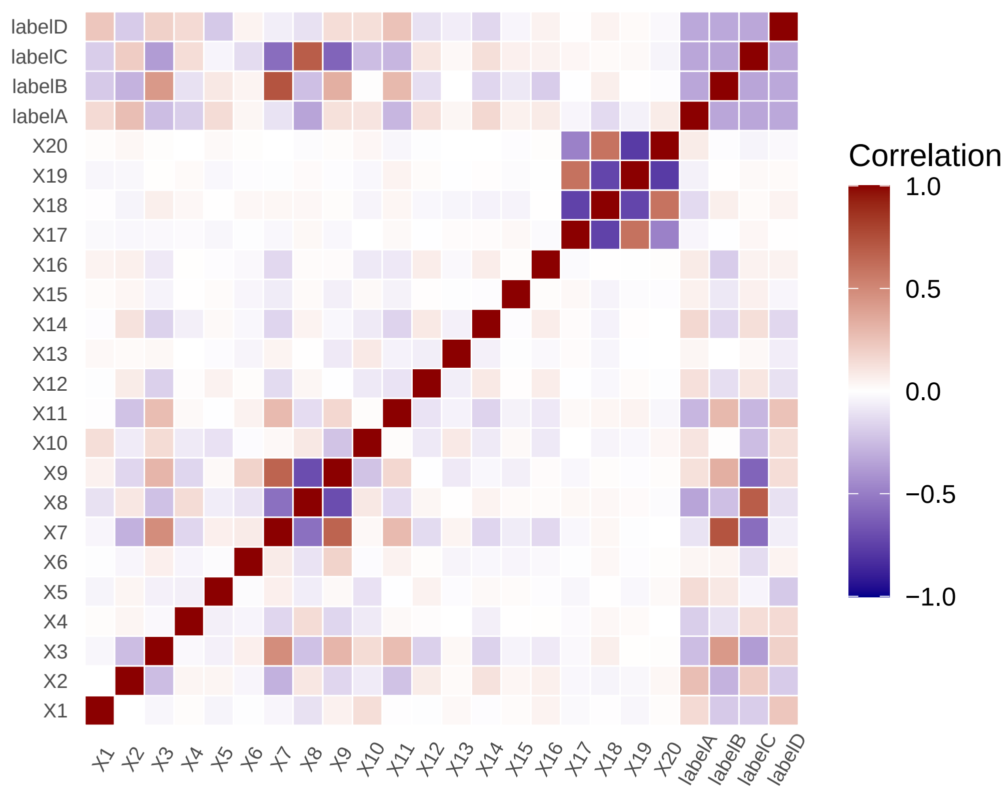



Figure 1: **Feature and Class Correlation Matrix:** *highlighting relationships between variables and relationships with catagorical labels*

Within the data there are two groups of features that correlate together - X7, X8, and X9, and X17 to X20. Noticing groups of correlated features is important since some models such as logistic regression and SVM will struggle with multicolinearity. This will lead us to attempt feature selection or dimensionality reduction with these models, or choose alternative algorithms that are more robust in these cases such as tree-based algorithms.

The difference between these two group is that while X7, X8, and X9 are three features with some of the strongest correlations with the label values, all of X17 to X20 are features without significant correlations. This would lead us to believe that X17 - X20 have low predictive power in classification that aim to predict the class label and so removing them entirely would be a justifyable approach.

Among the other columns, we see that there are definitly some columns with stronger correlations than others.

To produce the correlation matrix, the four catagorical labels were one hot encoded to create four binary columns. This allows us to see that several features have strong predictive power for one or more label but not all. For example, X8 has high correlations with classes A, B, and C, but none very low correlation with class D. This contrasts with a feature like X11 which has equal magnitude of correlation across all four labels.

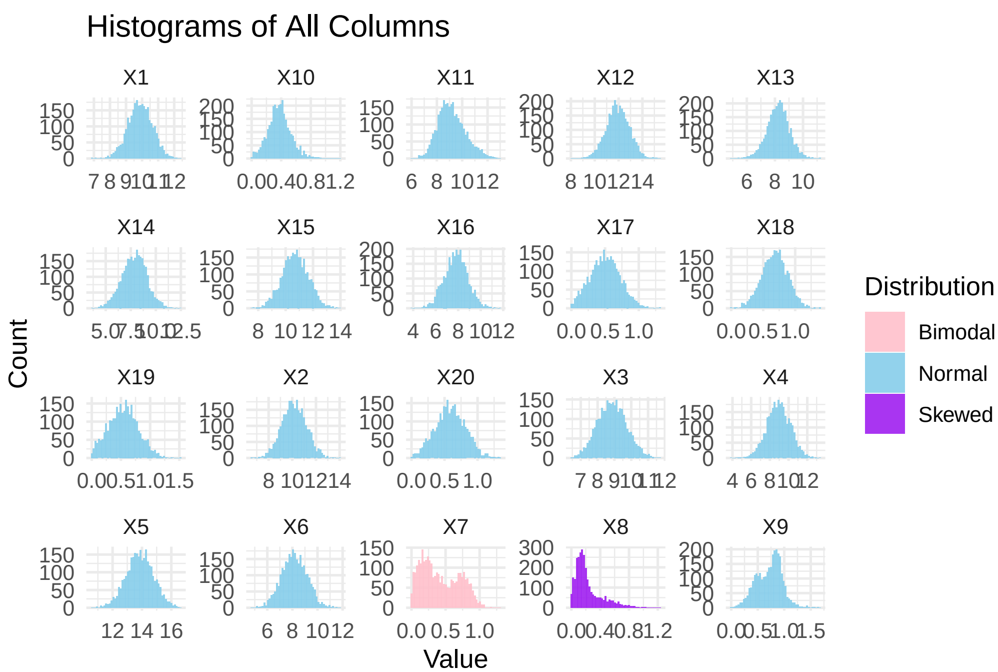



Figure 2: **Distributions within feature columns:** *histograms showing scale and skewness of data*

The majority of the twenty predictive variables appeared to follow a normal distribution. Noteable exceptions were X8, which is heavily right skewed, and X7 which has a bimodal appearance. With both of these features showing strong correlations with the labels leading to a high probability that they have strong predictive power, they should not be removed. X8 will be transformed, and the natural logarithm of X8 will be used in all modelling.

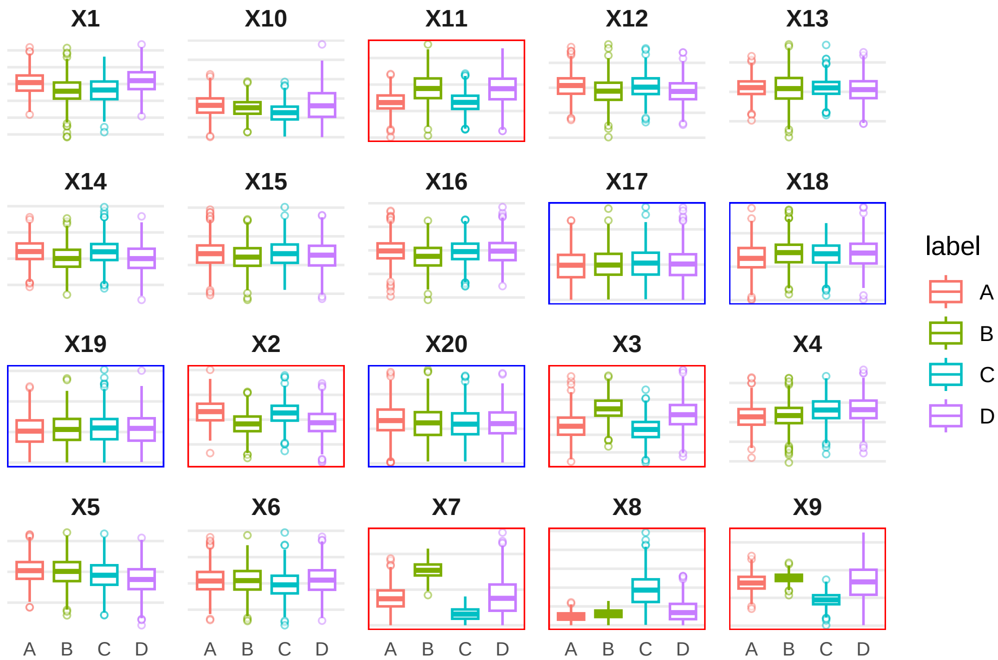



Figure 3: **Distributions within feature columns:** *Boxplots by class label by feature*

Across the entire dataset, the distribution of labels is uniform, with roughly on quarter of the observations falling into each category.

When the distribution of each feature is observed by class label, we see that some features have significantly different characteristics for each class. In the figure, highlighted with a red boarder are features where we can see notable differences in the key descriptive statistics such as medians and interquartile ranges between different classes. Highlighted in blue boarders are the features where the boxplots look almost identical from one feature to another. This gives us more insight into which features will be important for building effective models.

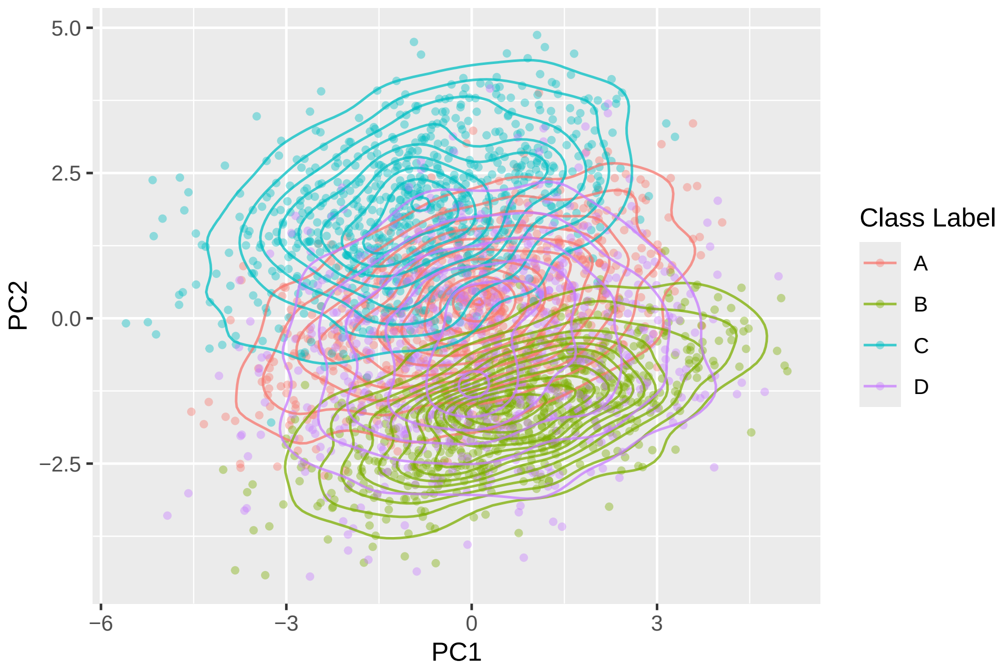



Figure 4: **PCA plotting of class labels:** *scatterplot showing clustering tendency of catagorical classes*

It is important that we learn about the structure of the underlying data in order to understand our chances of sucess with unsupervised methods where we do not have the value of the class label to train the model. Using Principle Componant Analysis, we can capture most of the information in the system in one scatter plot. This shows us that there is definitly some underlying structure to the data that will lead to the formation of clusters, although it is fairly loose in this plot. Annotating the points with the target label shows us that classes A, B, and C form elipsoidal groups that are roughly equal in size and orientation, and offset along the second principle component. The fourth class (D) forms a larger elipsoid that overlaps significantly with the other three.

The seperation of the first three classes suggests that there is information in the features that allow the statistical models to decern between them, but the overlap of class D means that this class may have more misclassifications. It may be challenging to find an approach that is effective for this label.

The fact that the three classes that are distinct are displaced along the second principle componant and not the first principle component means that it is the dimensions with less variation that distinguish them. It would be easier to seperate the groups with predictive models if they were offset along the first principle conmponent.

To learn more about the underlying structure of the dataset, the clustering tendency can be examined by calculating the Hopkins statistic, where a score close to 1 indicates strong clustering tenency, a score of 0.5 indicates random distribution of occurences, and 0 a uniform distribution.

Table 2: Hopkins Statistic Scores

| Columns used | Hopkins Statistic Score |
|---|---|
| All Features + Label | 0.9999689 |
| All Features with Label Removed | 0.9999401 |
| All Features Binary Class D vs Rest | 0.9999687 |
| Features X2,X3,X7,X8,X9,X11 | 0.9943937 |
| Features X17,X18,X19,X20 | 0.9982655 |
| Features X1,X4,X6,X10,X12,X13,X14,X15,X16 | 0.9962138 |

We can see there is a very high clustering tendency for various treatments of the data. The statistic remains high when the label is removed from the feature set. This is promising for persuing unsupervised approaches, since it confirms that the 20 numeric features contain the information that are structuring the data into clusters, rather than the label itself providing a significant amount of this structure. If we saw the score drop when the label was removed, this would suggest that the label was necessary for dividing the data into classes and the features themselves did not contain sufficient predictive information to do so.

Similarly, we see that the score remains high for three different groups of features. These three groups are the groups we see with different coloured boarders in the boxplots. This means that even the features with very little correlation with the labels provide structure to the data. This could suggest that there are other ways that the data could be structured when using unsupervised models.

Overall, we see from exploritory data analysis that this data contains high amounts of information usable for predictive modelling, and a high clustering tendency which is promising for unsupervised clustering. Multicolinarity has been observed among several features that may prevent models from performing well.

## 2.3 Supervised Learning Methods

### 2.3.1 logistic regression - including class weighting and L2 regularization, and feature selection.

A simple model, logistic regression is quick to implement and will reveal more about the data, in particular how different features contribute to predictions.

There are variations such as weighting and regularization which will be implemented - the success of lack of success of these techniques will reveal characteristics about our data that will be valuable to inform the selection and implementation of other models

### 2.3.2 Random forest, including feature selection and tuning of mtry.

Random Forest was chosen due to its resilience. One of the more robust option, it is a good choice for handling the data without extra processing. Several characteristics of the data have been identified that may cause othe models such as logistic regression and SVM to struggle: - Correlated of features -Features that appeared to have low linear correlations with the label values (from heatmap in eda) but still contributed to the predictive ability of the model. This suggests there might be some non-linear relationships between features and the target variable - Features that aren't perfectly normally distributed, such as the bimodal peak in X7

Random Forest is a robust algorithm with few underlying assumptions that will handle these considerations well. Random Forest resists overfitting because of the sampling approach, it handles non linearity well, and it is naturally suited to multinomial classifications problems, like the one we have with four possible values for label. I also think that tree based models may peform well at distinguishing class A from class D, which was the biggest challenge that held back our logistic regression modelling. This is because it can prioritize at an early node in the tree a feature such as X11, which is one of the few that had high importance for descerning between class A and D, and then refine the selection in further nodes.

### 2.3.3 SVM, with feature selection and tuning of kernel selection, gamma and cost parameters.

SVM is a powerful and popular algorith. SVM has options for different kernals that can be tuned, and this is a promising approach to solving the challenges of seperating class A from class D that is evedent from the data analysis and the results of logistic regression. It might be the case that A and D aren't linearly seperable, but a non-linear kernal will have success.

## 2.4 Unsupervised Learning Methods

### 2.4.1 Agglomerative hierarchical clustering, including tuning of linking metric.

Agglomerative heirarchical clustering was chosen because it is interesting to explore a model where the number of clusters is not specified and let the natural structure of the data reveal itself.

Biological data is often naturally heirarchical, for example animals can be classified by deviding them into first kingdoms, then families of species, and finally species and sub-species.

Although we don't know the exact meaning of each feature in our data, we know that it is biological in origin, perhaps gene expressions or environmental factors. This means that there might be a heirarchy of classes in our data.

Using this method without specifying that there are four values for label might reveal that some of the labels have a strong tendency to form sub classes, or that there is little structure in the data to justify asserting there are four classes. These would both be interesting finds.

It is also a model that handles the feature correlation well, which allows us to keep in all the collumns that correlate like X7, X8, and X9.

#### 2.4.2 Gaussian mixture model based clustering, including selecting a model, regularization using shrinkage parameter, and dimensionality reduction using factor analysis.

So far while working with this data set, we have struggled to seperate class D, and by plotting the results of some of the methods in specific dimensions, we have been able to show that class D significantly overlaps the other classes. We have also seen from the two dimensional PCA scatter-plot that this is general overlap between all the classes in the the first two principle componants of the data.

Lots of clustering methods struggle with seperating overlapping clusters, so for the final method I wanted to choose one that might perform better with this challenge in mind. I have chosen to try a gaussian mixture model - a model-based clustering technique that assumes that all the data is distributed according to the combination of different normal distributions. I think it might have a good chance of performing well on our dataset because it is probabalistic, calculating the probability that a data point is in each cluster. This can help it perform better than other methods like K-means when there aren't clear boundaries between the clusters such as we see with class D.

Another advantage it has is that it has some flexibility in the geometry of the clusters it produces, unlike k-means which tends to produce spherical clusters. This is important because we have seen that in some dimensions our classes produce fairly elipsoidal clusters. It also operates on very different fundemental principles to our other unsupervised method - agglomerative heirarchical clustering - so it will be good to compare the two. If model-based clustering performs much better then it could suggest that the classes of our data are not heirarchical in nature.

## 3 Results

### 3.1 Exploratory Data Analysis Findings

most features are normally distributed except for X8 which is highly skewed. Features originally had different scales. X7, X8, X9 columns are correlated and correlate highly with the labels. X17, X18, X19 and X20 are highly correlated together and have very low correlation with the labels.

The PCA showed the four labels had some clustered structure, but also some significant overlap. The hopkins statistic showed there was a moderate clustering tendency, but that the label column when included made the clustering tendency extremely high. This is an initial suggestion that supervised learning would be more effective than unsupervised learning

## 3.2 Supervised Learning Results

### 3.2.1 Logistic Regression

Table 3: Simple Logistic Regression Confusion Matrix

| | Reference | | | |
|---|---|---|---|---|
| Prediction | A | B | C | D |
| A | 128 | 1 | 3 | 14 |
| B | 0 | 129 | 0 | 20 |
| C | 0 | 0 | 140 | 17 |
| D | 12 | 8 | 4 | 77 |

Table 4: Simple Logistic Regression Overall Statistics

| | Statistic | Value |
|---|---|---|
| Accuracy | Accuracy | 0.8571 |
| Kappa | Kappa | 0.8091 |
| AccuracyLower | AccuracyLower | 0.8252 |
| AccuracyUpper | AccuracyUpper | 0.8852 |
| AccuracyNull | AccuracyNull | 0.2658 |
| AccuracyPValue | AccuracyPValue | 0.0000 |
| McnemarPValue | McnemarPValue | NaN |

Table 5: Simple Logistic Regression Statistics by Class

| Statistic | Class: A | Class: B | Class: C | Class: D |
|---|---|---|---|---|
| Sensitivity | 0.9143 | 0.9348 | 0.9524 | 0.6016 |
| Specificity | 0.9564 | 0.9518 | 0.9581 | 0.9435 |
| Pos Pred Value | 0.8767 | 0.8658 | 0.8917 | 0.7624 |
| Neg Pred Value | 0.9705 | 0.9777 | 0.9823 | 0.8872 |
| Precision | 0.8767 | 0.8658 | 0.8917 | 0.7624 |
| Recall | 0.9143 | 0.9348 | 0.9524 | 0.6016 |
| F1 | 0.8951 | 0.8990 | 0.9211 | 0.6725 |
| Prevalence | 0.2532 | 0.2495 | 0.2658 | 0.2315 |
| Detection Rate | 0.2315 | 0.2333 | 0.2532 | 0.1392 |
| Detection Prevalence | 0.2640 | 0.2694 | 0.2839 | 0.1826 |
| Balanced Accuracy | 0.9354 | 0.9433 | 0.9553 | 0.7725 |

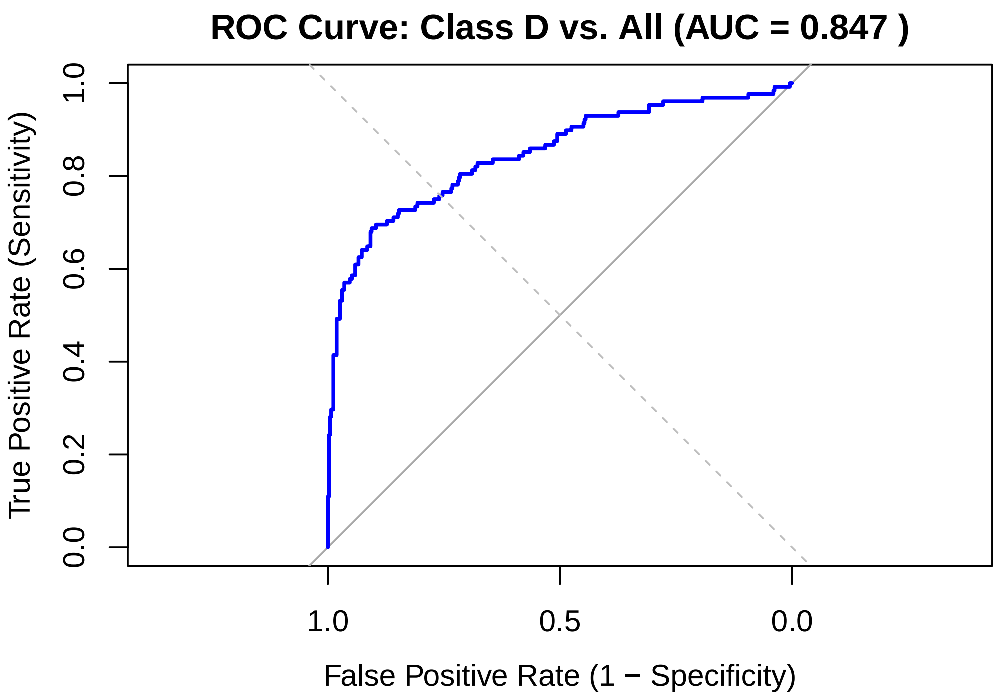



Figure 5: **Simple Logistic Regression ROC Plot:** *ROC plot for Class D vs Not Class D*

Table 6: Simple Logistic Regression with selected features (X17,X19,X20 removed) Confusion Matrix

| | Reference | | | |
|---|---|---|---|---|
| Prediction | A | B | C | D |
| A | 122 | 1 | 4 | 16 |
| B | 0 | 128 | 0 | 18 |
| C | 1 | 0 | 141 | 19 |
| D | 17 | 9 | 2 | 75 |

Table 7: Simple Logistic Regression with selected features (X17,X19,X20 removed) Overall Statistics

| | Statistic | Value |
|---|---|---|
| Accuracy | Accuracy | 0.8427 |
| Kappa | Kappa | 0.7897 |
| AccuracyLower | AccuracyLower | 0.8096 |
| AccuracyUpper | AccuracyUpper | 0.8720 |
| AccuracyNull | AccuracyNull | 0.2658 |
| AccuracyPValue | AccuracyPValue | 0.0000 |
| McnemarPValue | McnemarPValue | NaN |

Table 8: Simple Logistic Regression with selected features (X17,X19,X20 removed) Statistics by Class

| Statistic | Class: A | Class: B | Class: C | Class: D |
|---|---|---|---|---|
| Sensitivity | 0.8714 | 0.9275 | 0.9592 | 0.5859 |
| Specificity | 0.9492 | 0.9566 | 0.9507 | 0.9341 |
| Pos Pred Value | 0.8531 | 0.8767 | 0.8758 | 0.7282 |
| Neg Pred Value | 0.9561 | 0.9754 | 0.9847 | 0.8822 |
| Precision | 0.8531 | 0.8767 | 0.8758 | 0.7282 |
| Recall | 0.8714 | 0.9275 | 0.9592 | 0.5859 |
| F1 | 0.8622 | 0.9014 | 0.9156 | 0.6494 |
| Prevalence | 0.2532 | 0.2495 | 0.2658 | 0.2315 |
| Detection Rate | 0.2206 | 0.2315 | 0.2550 | 0.1356 |
| Detection Prevalence | 0.2586 | 0.2640 | 0.2911 | 0.1863 |
| Balanced Accuracy | 0.9103 | 0.9421 | 0.9550 | 0.7600 |

Table 9: Simple Logistic Regression with selected features (X8,X9 removed) Confusion Matrix

| | Reference | | | |
|---|---|---|---|---|
| Prediction | A | B | C | D |
| A | 112 | 3 | 8 | 16 |
| B | 4 | 119 | 0 | 14 |
| C | 14 | 0 | 127 | 28 |
| D | 10 | 16 | 12 | 70 |

Table 10: Simple Logistic Regression with selected features (X8,X9 removed) Overall Statistics

| | Statistic | Value |
|---|---|---|
| Accuracy | Accuracy | 0.7740 |
| Kappa | Kappa | 0.6978 |
| AccuracyLower | AccuracyLower | 0.7368 |
| AccuracyUpper | AccuracyUpper | 0.8082 |
| AccuracyNull | AccuracyNull | 0.2658 |
| AccuracyPValue | AccuracyPValue | 0.0000 |
| McnemarPValue | McnemarPValue | NaN |

Table 11: Simple Logistic Regression with selected features (X8,X9 removed) Statistics by Class

| Statistic | Class: A | Class: B | Class: C | Class: D |
|---|---|---|---|---|
| Sensitivity | 0.8000 | 0.8623 | 0.8639 | 0.5469 |
| Specificity | 0.9346 | 0.9566 | 0.8966 | 0.9106 |
| Pos Pred Value | 0.8058 | 0.8686 | 0.7515 | 0.6481 |
| Neg Pred Value | 0.9324 | 0.9543 | 0.9479 | 0.8697 |
| Precision | 0.8058 | 0.8686 | 0.7515 | 0.6481 |
| Recall | 0.8000 | 0.8623 | 0.8639 | 0.5469 |

| | | | | |
|---|---|---|---|---|
| F1 | 0.8029 | 0.8655 | 0.8038 | 0.5932 |
| Prevalence | 0.2532 | 0.2495 | 0.2658 | 0.2315 |
| Detection Rate | 0.2025 | 0.2152 | 0.2297 | 0.1266 |
| Detection Prevalence | 0.2514 | 0.2477 | 0.3056 | 0.1953 |
| Balanced Accuracy | 0.8673 | 0.9095 | 0.8802 | 0.7287 |

Table 12: Weighted Logistic Regression Confusion Matrix

| | Reference | | | |
|---|---|---|---|---|
| Prediction | A | B | C | D |
| A | 128 | 1 | 3 | 13 |
| B | 0 | 128 | 0 | 19 |
| C | 0 | 0 | 139 | 16 |
| D | 12 | 9 | 5 | 80 |

Table 13: Weighted Logistic Regression Overall Statistics

| | Statistic | Value |
|---|---|---|
| Accuracy | Accuracy | 0.8571 |
| Kappa | Kappa | 0.8091 |
| AccuracyLower | AccuracyLower | 0.8252 |
| AccuracyUpper | AccuracyUpper | 0.8852 |
| AccuracyNull | AccuracyNull | 0.2658 |
| AccuracyPValue | AccuracyPValue | 0.0000 |
| McnemarPValue | McnemarPValue | NaN |

Table 14: Weighted Logistic Regression Statistics by Class

| Statistic | Class: A | Class: B | Class: C | Class: D |
|---|---|---|---|---|
| Sensitivity | 0.9143 | 0.9275 | 0.9456 | 0.6250 |
| Specificity | 0.9588 | 0.9542 | 0.9606 | 0.9388 |
| Pos Pred Value | 0.8828 | 0.8707 | 0.8968 | 0.7547 |
| Neg Pred Value | 0.9706 | 0.9754 | 0.9799 | 0.8926 |
| Precision | 0.8828 | 0.8707 | 0.8968 | 0.7547 |
| Recall | 0.9143 | 0.9275 | 0.9456 | 0.6250 |
| F1 | 0.8982 | 0.8982 | 0.9205 | 0.6838 |
| Prevalence | 0.2532 | 0.2495 | 0.2658 | 0.2315 |
| Detection Rate | 0.2315 | 0.2315 | 0.2514 | 0.1447 |
| Detection Prevalence | 0.2622 | 0.2658 | 0.2803 | 0.1917 |
| Balanced Accuracy | 0.9366 | 0.9409 | 0.9531 | 0.7819 |

Table 15: Logistic Regression Performance with Different Regularization Parameters

| Decay | Accuracy | Kappa | F1 Score (A) | F1 Score (B) | F1 Score (C) | F1 Score (D) |
|---|---|---|---|---|---|---|
| 0.001 | 0.8571 | 0.8091 | 0.8951 | 0.8990 | 0.9211 | 0.6725 |
| 0.01 | 0.8571 | 0.8091 | 0.8951 | 0.8990 | 0.9211 | 0.6725 |
| 0.1 | 0.8571 | 0.8091 | 0.8951 | 0.8990 | 0.9211 | 0.6725 |
| 0.5 | 0.8590 | 0.8115 | 0.9021 | 0.8951 | 0.9216 | 0.6754 |
| 1 | 0.8571 | 0.8090 | 0.9021 | 0.8920 | 0.9216 | 0.6696 |
| 2 | 0.8571 | 0.8090 | 0.8951 | 0.9028 | 0.9186 | 0.6667 |
| 10 | 0.8481 | 0.7969 | 0.8723 | 0.9059 | 0.9097 | 0.6608 |

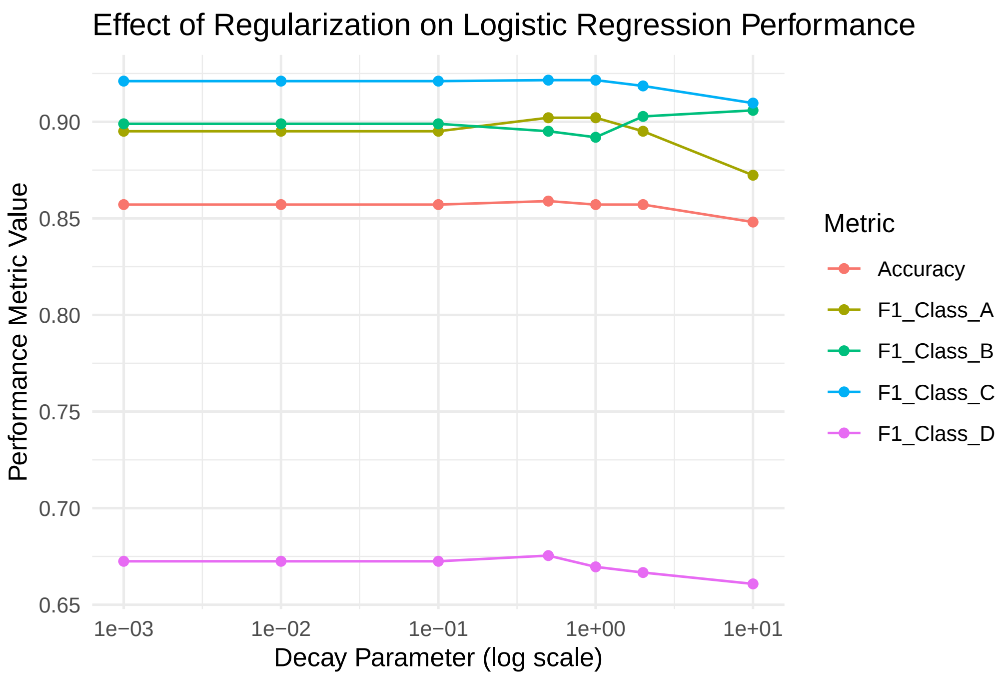



Figure 6: **Regularized Logistic Regression**

Table 16: Confusion Matrix for Best Regularized Logistic Regression Model (Decay = 0.5 )

| | Reference | | | |
|---|---|---|---|---|
| Prediction | A | B | C | D |
| A | 129 | 1 | 3 | 13 |
| B | 0 | 128 | 0 | 20 |
| C | 0 | 0 | 141 | 18 |
| D | 11 | 9 | 3 | 77 |

logistic regression was not great. weighting did nothing, as expected. regularization didn't really do anything. feature selection did not improve the model. We saw that the model particularly underperformed at classifying class D correctly.

### 3.2.2 Random Forest

Table 17: Basic Random Forest Confusion Matrix

| | Reference | | | |
|---|---|---|---|---|
| Prediction | A | B | C | D |
| A | 131 | 1 | 1 | 17 |
| B | 1 | 135 | 0 | 9 |
| C | 1 | 0 | 144 | 12 |
| D | 7 | 2 | 2 | 90 |

Table 18: Basic Random Forest Overall Statistics

| | Statistic | Value |
|---|---|---|
| Accuracy | Accuracy | 0.9042 |
| Kappa | Kappa | 0.8719 |
| AccuracyLower | AccuracyLower | 0.8765 |
| AccuracyUpper | AccuracyUpper | 0.9274 |
| AccuracyNull | AccuracyNull | 0.2658 |
| AccuracyPValue | AccuracyPValue | 0.0000 |
| McnemarPValue | McnemarPValue | NaN |

Table 19: Basic Random Forest Statistics by Class

| Statistic | Class: A | Class: B | Class: C | Class: D |
|---|---|---|---|---|
| Sensitivity | 0.9357 | 0.9783 | 0.9796 | 0.7031 |
| Specificity | 0.9540 | 0.9759 | 0.9680 | 0.9741 |
| Pos Pred Value | 0.8733 | 0.9310 | 0.9172 | 0.8911 |
| Neg Pred Value | 0.9777 | 0.9926 | 0.9924 | 0.9159 |
| Precision | 0.8733 | 0.9310 | 0.9172 | 0.8911 |
| Recall | 0.9357 | 0.9783 | 0.9796 | 0.7031 |
| F1 | 0.9034 | 0.9541 | 0.9474 | 0.7860 |
| Prevalence | 0.2532 | 0.2495 | 0.2658 | 0.2315 |
| Detection Rate | 0.2369 | 0.2441 | 0.2604 | 0.1627 |
| Detection Prevalence | 0.2712 | 0.2622 | 0.2839 | 0.1826 |
| Balanced Accuracy | 0.9449 | 0.9771 | 0.9738 | 0.8386 |

Table 20: Basic Random Forest Feature Importance

| | A | B | C | D | MeanDecreaseAccuracy | MeanDecreaseGini |
|---|---|---|---|---|---|---|
| X7 | 56.558 | 118.667 | 49.213 | 12.349 | 95.167 | 359.608 |
| X8 | 65.991 | 25.980 | 62.515 | -6.441 | 69.507 | 251.828 |
| X10 | 46.550 | 24.751 | 52.066 | 21.225 | 60.814 | 132.552 |
| X9 | 36.974 | 30.870 | 52.227 | 14.993 | 57.843 | 241.034 |
| X11 | 41.356 | 11.814 | 39.916 | 29.586 | 52.587 | 113.984 |

| | | | | | | |
|---|---|---|---|---|---|---|
| X3 | 27.784 | 27.346 | 32.775 | 14.129 | 44.176 | 108.661 |
| X2 | 19.998 | 13.586 | 13.090 | 8.573 | 25.746 | 62.498 |
| X1 | 8.705 | 13.112 | 11.541 | 15.451 | 22.705 | 54.390 |
| X4 | 14.772 | 4.246 | 2.835 | 7.697 | 16.201 | 40.565 |
| X5 | 13.007 | 3.643 | 0.538 | 12.547 | 15.756 | 38.826 |
| X14 | 10.934 | 1.717 | 7.500 | 5.310 | 11.994 | 36.633 |
| X13 | 6.940 | -0.114 | 6.335 | 7.238 | 10.488 | 30.080 |
| X18 | 10.786 | 4.458 | 3.321 | 0.596 | 9.840 | 27.809 |
| X12 | 4.842 | -0.140 | 5.828 | 5.614 | 8.871 | 28.373 |
| X20 | 6.059 | 3.517 | 3.691 | -0.660 | 6.853 | 21.973 |
| X19 | 3.013 | 4.775 | 1.034 | -0.136 | 4.629 | 20.874 |
| X16 | 4.336 | 2.854 | 0.245 | 0.702 | 4.116 | 26.808 |
| X15 | 1.161 | 1.232 | 3.272 | -0.097 | 2.703 | 23.289 |
| X17 | 2.575 | 4.210 | 0.048 | -1.737 | 2.649 | 20.824 |
| X6 | -0.912 | 1.116 | 3.538 | 0.351 | 1.909 | 23.800 |

Table 21: Random Forest (5 least important features removed) Confusion Matrix

| | Reference | | | |
|---|---|---|---|---|
| Prediction | A | B | C | D |
| A | 130 | 1 | 2 | 19 |
| B | 1 | 134 | 0 | 9 |
| C | 1 | 0 | 142 | 11 |
| D | 8 | 3 | 3 | 89 |

Table 22: Random Forest (5 least important features removed) Overall Statistics

| | Statistic | Value |
|---|---|---|
| Accuracy | Accuracy | 0.8951 |
| Kappa | Kappa | 0.8598 |
| AccuracyLower | AccuracyLower | 0.8665 |
| AccuracyUpper | AccuracyUpper | 0.9194 |
| AccuracyNull | AccuracyNull | 0.2658 |
| AccuracyPValue | AccuracyPValue | 0.0000 |
| McnemarPValue | McnemarPValue | NaN |

Table 23: Random Forest (5 least important features removed) Statistics by Class

| Statistic | Class: A | Class: B | Class: C | Class: D |
|---|---|---|---|---|
| Sensitivity | 0.9286 | 0.9710 | 0.9660 | 0.6953 |
| Specificity | 0.9467 | 0.9759 | 0.9704 | 0.9671 |
| Pos Pred Value | 0.8553 | 0.9306 | 0.9221 | 0.8641 |
| Neg Pred Value | 0.9751 | 0.9902 | 0.9875 | 0.9133 |
| Precision | 0.8553 | 0.9306 | 0.9221 | 0.8641 |
| Recall | 0.9286 | 0.9710 | 0.9660 | 0.6953 |
| F1 | 0.8904 | 0.9504 | 0.9435 | 0.7706 |

| | | | | |
|---|---|---|---|---|
| Prevalence | 0.2532 | 0.2495 | 0.2658 | 0.2315 |
| Detection Rate | 0.2351 | 0.2423 | 0.2568 | 0.1609 |
| Detection Prevalence | 0.2749 | 0.2604 | 0.2785 | 0.1863 |
| Balanced Accuracy | 0.9377 | 0.9735 | 0.9682 | 0.8312 |

Table 24: Random Forest (5 least important features removed) Feature Importance

| | A | B | C | D | MeanDecreaseAccuracy | MeanDecreaseGini |
|---|---|---|---|---|---|---|
| X7 | 61.055 | 177.557 | 55.712 | 14.925 | 113.337 | 397.720 |
| X8 | 88.792 | 25.710 | 73.925 | -7.703 | 87.006 | 278.636 |
| X10 | 55.157 | 28.584 | 65.397 | 21.244 | 78.331 | 153.548 |
| X11 | 49.151 | 11.713 | 46.621 | 30.848 | 64.485 | 124.809 |
| X9 | 37.606 | 31.477 | 56.004 | 17.270 | 64.405 | 251.544 |
| X3 | 26.990 | 27.271 | 31.846 | 13.963 | 45.209 | 109.174 |
| X2 | 21.793 | 13.329 | 13.399 | 9.146 | 25.922 | 60.077 |
| X1 | 8.169 | 13.426 | 13.781 | 15.236 | 23.701 | 52.354 |
| X4 | 17.311 | 5.883 | -0.227 | 9.381 | 18.843 | 41.735 |
| X5 | 13.363 | 2.294 | -0.812 | 12.958 | 16.715 | 39.172 |
| X14 | 10.922 | -0.500 | 6.611 | 6.240 | 12.783 | 37.323 |
| X13 | 7.942 | 0.018 | 8.406 | 4.494 | 10.923 | 30.103 |
| X12 | 4.915 | -0.774 | 6.641 | 6.986 | 9.581 | 30.379 |
| X18 | 7.242 | -1.901 | 4.388 | 2.293 | 6.436 | 29.636 |
| X16 | 4.754 | 2.667 | 0.493 | 2.047 | 4.961 | 28.217 |

Table 25: Random Forest (5 least important features removed) Confusion Matrix

| | Reference | | | |
|---|---|---|---|---|
| Prediction | A | B | C | D |
| A | 128 | 2 | 1 | 19 |
| B | 2 | 134 | 0 | 8 |
| C | 2 | 0 | 143 | 12 |
| D | 8 | 2 | 3 | 89 |

Table 26: Random Forest (5 least important features removed) Overall Statistics

| | Statistic | Value |
|---|---|---|
| Accuracy | Accuracy | 0.8933 |
| Kappa | Kappa | 0.8574 |
| AccuracyLower | AccuracyLower | 0.8645 |
| AccuracyUpper | AccuracyUpper | 0.9178 |
| AccuracyNull | AccuracyNull | 0.2658 |
| AccuracyPValue | AccuracyPValue | 0.0000 |
| McnemarPValue | McnemarPValue | NaN |

Table 27: Random Forest (5 least important features removed) Statistics by Class

| Statistic | Class: A | Class: B | Class: C | Class: D |
|---|---|---|---|---|
| Sensitivity | 0.9143 | 0.9710 | 0.9728 | 0.6953 |
| Specificity | 0.9467 | 0.9759 | 0.9655 | 0.9694 |
| Pos Pred Value | 0.8533 | 0.9306 | 0.9108 | 0.8725 |
| Neg Pred Value | 0.9702 | 0.9902 | 0.9899 | 0.9135 |
| Precision | 0.8533 | 0.9306 | 0.9108 | 0.8725 |
| Recall | 0.9143 | 0.9710 | 0.9728 | 0.6953 |
| F1 | 0.8828 | 0.9504 | 0.9408 | 0.7739 |
| Prevalence | 0.2532 | 0.2495 | 0.2658 | 0.2315 |
| Detection Rate | 0.2315 | 0.2423 | 0.2586 | 0.1609 |
| Detection Prevalence | 0.2712 | 0.2604 | 0.2839 | 0.1844 |
| Balanced Accuracy | 0.9305 | 0.9735 | 0.9692 | 0.8324 |

Table 28: Random Forest (5 least important features removed) Feature Importance

| | A | B | C | D | MeanDecreaseAccuracy | MeanDecreaseGini |
|---|---|---|---|---|---|---|
| X7 | 49.378 | 124.698 | 53.309 | 12.102 | 97.825 | 384.170 |
| X9 | 47.157 | 35.396 | 111.489 | 16.624 | 87.840 | 334.826 |
| X10 | 52.393 | 24.362 | 55.209 | 21.704 | 66.744 | 146.548 |
| X11 | 42.960 | 13.588 | 43.828 | 31.389 | 59.129 | 123.378 |
| X3 | 26.132 | 36.587 | 35.320 | 14.353 | 50.914 | 123.061 |
| X2 | 21.464 | 14.659 | 15.439 | 10.549 | 29.079 | 71.440 |
| X1 | 9.646 | 14.493 | 15.000 | 15.215 | 26.767 | 66.584 |
| X4 | 13.486 | 4.010 | 4.321 | 10.485 | 17.401 | 49.500 |
| X5 | 12.570 | 3.156 | -0.769 | 14.931 | 16.060 | 45.017 |
| X14 | 11.881 | 2.210 | 7.208 | 9.348 | 14.554 | 43.894 |
| X18 | 11.369 | 5.195 | 5.493 | 2.827 | 12.462 | 34.427 |
| X13 | 6.973 | -0.130 | 6.648 | 4.022 | 8.522 | 35.437 |
| X12 | 2.292 | 0.042 | 8.676 | 4.968 | 8.161 | 34.572 |
| X16 | 3.370 | 4.143 | 2.393 | 5.357 | 7.315 | 33.664 |
| X20 | 5.940 | 1.541 | 4.773 | 0.424 | 6.834 | 27.432 |
| X19 | 4.392 | 3.560 | 4.815 | 0.037 | 6.577 | 26.463 |
| X17 | 2.796 | 4.223 | 3.016 | 1.059 | 5.315 | 25.539 |
| X6 | 1.709 | 0.087 | 4.610 | 0.832 | 3.575 | 30.082 |
| X15 | 0.840 | 1.282 | 4.396 | 0.137 | 3.127 | 28.455 |

```
-0.04366812 0.01
-0.004366812 0.01
```

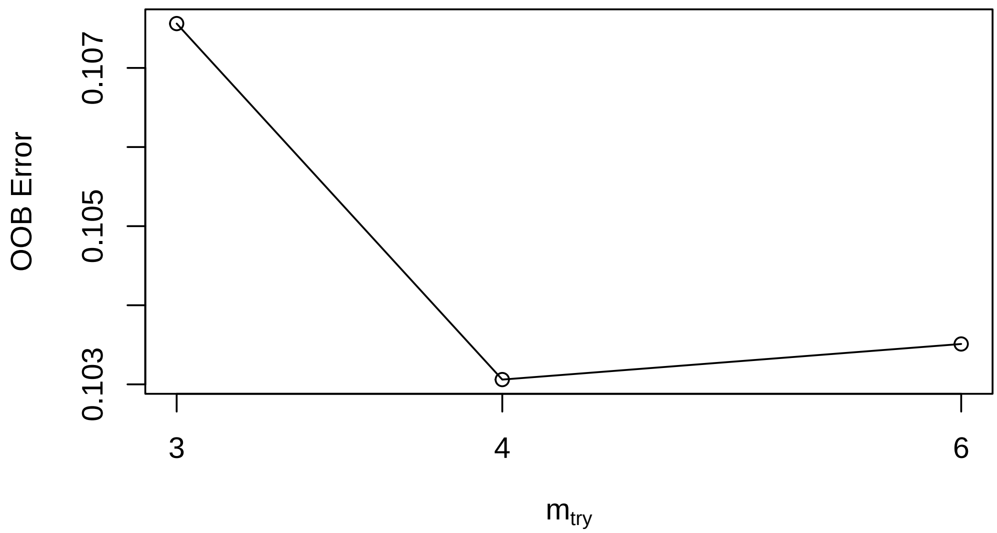



Figure 7: **Optimal mtry for Random Forest**

Table 29: Tuned Random Forest Confusion Matrix

| | Reference | | | |
|---|---|---|---|---|
| Prediction | A | B | C | D |
| A | 129 | 1 | 2 | 18 |
| B | 2 | 134 | 0 | 9 |
| C | 1 | 0 | 141 | 8 |
| D | 8 | 3 | 4 | 93 |

Table 30: Tuned Random Forest Overall Statistics

| | Statistic | Value |
|---|---|---|
| Accuracy | Accuracy | 0.9042 |
| Kappa | Kappa | 0.8719 |
| AccuracyLower | AccuracyLower | 0.8765 |
| AccuracyUpper | AccuracyUpper | 0.9274 |
| AccuracyNull | AccuracyNull | 0.2658 |
| AccuracyPValue | AccuracyPValue | 0.0000 |
| McnemarPValue | McnemarPValue | NaN |

Table 31: Tuned Random Forest Statistics by Class

| Statistic | Class: A | Class: B | Class: C | Class: D |
|---|---|---|---|---|
| Sensitivity | 0.9214 | 0.9710 | 0.9592 | 0.7266 |
| Specificity | 0.9492 | 0.9735 | 0.9778 | 0.9647 |
| Pos Pred Value | 0.8600 | 0.9241 | 0.9400 | 0.8611 |
| Neg Pred Value | 0.9727 | 0.9902 | 0.9851 | 0.9213 |

| | | | | |
|---|---|---|---|---|
| Precision | 0.8600 | 0.9241 | 0.9400 | 0.8611 |
| Recall | 0.9214 | 0.9710 | 0.9592 | 0.7266 |
| F1 | 0.8897 | 0.9470 | 0.9495 | 0.7881 |
| Prevalence | 0.2532 | 0.2495 | 0.2658 | 0.2315 |
| Detection Rate | 0.2333 | 0.2423 | 0.2550 | 0.1682 |
| Detection Prevalence | 0.2712 | 0.2622 | 0.2712 | 0.1953 |
| Balanced Accuracy | 0.9353 | 0.9723 | 0.9685 | 0.8456 |

The random forest performed well without any configuration. feature selection was not effective. still struggled at seperating class D. mtry was tuned.

### 3.2.3 SVM

Table 32: SVM Performance with Different Kernels

| Kernel | Accuracy | Kappa | F1 Score (A) | F1 Score (B) | F1 Score (C) | F1 Score (D) |
|---|---|---|---|---|---|---|
| Linear | 0.8662 | 0.8212 | 0.8990 | 0.9155 | 0.9211 | 0.6926 |
| Polynomial - Order 3 | 0.8463 | 0.7944 | 0.8562 | 0.9143 | 0.9145 | 0.6481 |
| Polynomial - Order 5 | 0.7324 | 0.6415 | 0.6974 | 0.8276 | 0.8159 | 0.5393 |
| Polynomial - Order 7 | 0.5371 | 0.3797 | 0.5404 | 0.5810 | 0.6271 | 0.3268 |
| Radial | 0.8843 | 0.8452 | 0.9048 | 0.9371 | 0.9346 | 0.7182 |
| Sigmoid | 0.7125 | 0.6159 | 0.7260 | 0.7823 | 0.8383 | 0.4583 |

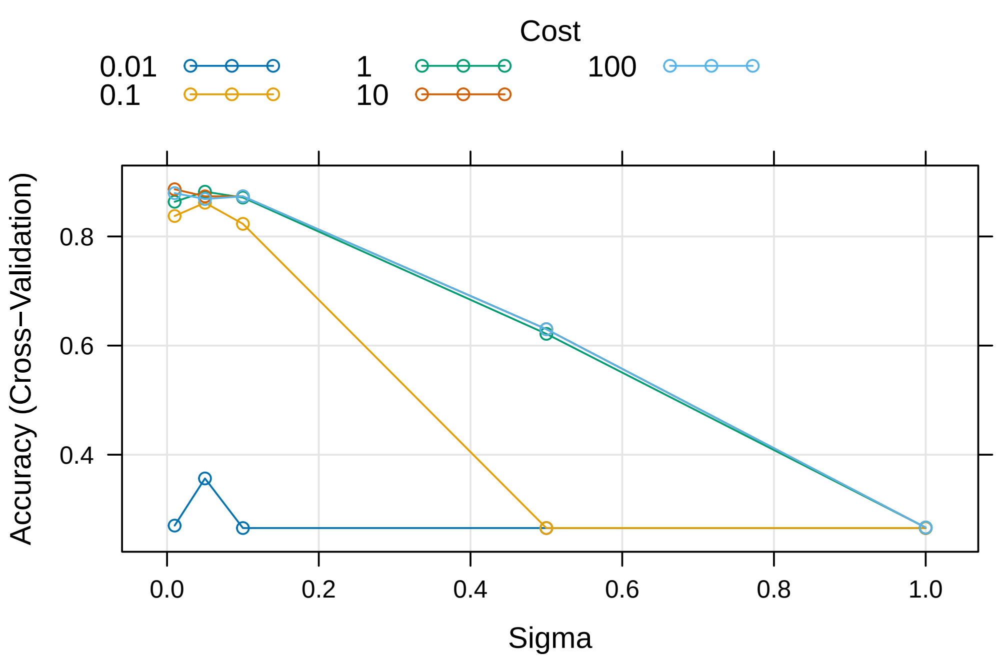



Figure 8: **Hyperparameter Affect on SVM Performance**

svm was good but not as good as random forest. lots of tuning. feature selection was uneffective

## 3.3 Unsupervised Learning Results

### 3.3.1 Agglomerative Hierarchical Clustering

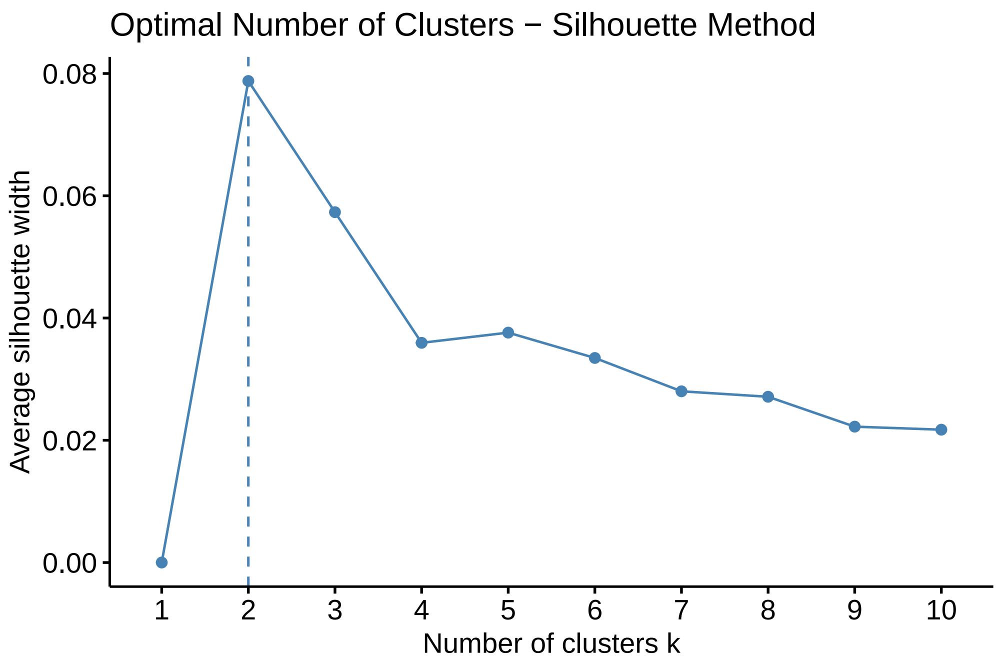



Figure 9: **Silhouette Plot**

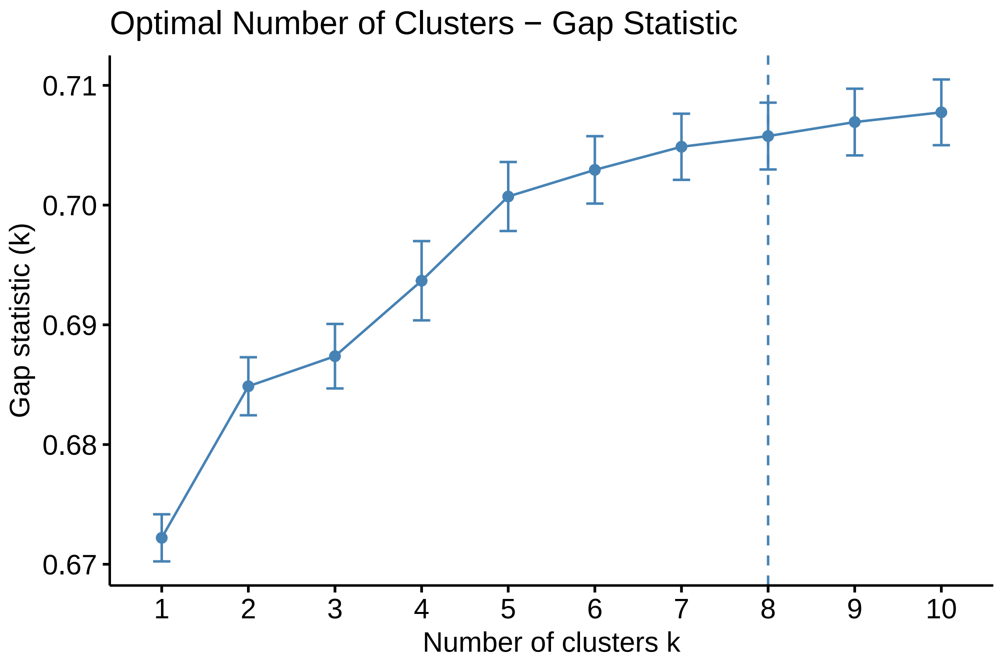



Figure 10: **Gap Statistic Plot**

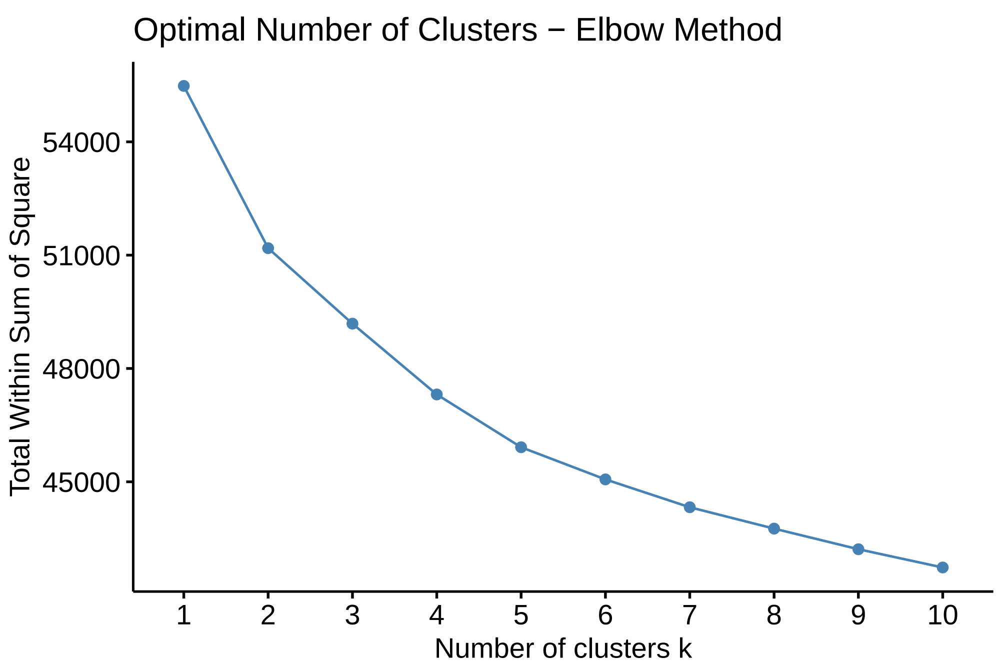



Figure 11: **Elbow Graph**

ahc was good not great.

### 3.3.2 Gaussian Mixed Model Clustering

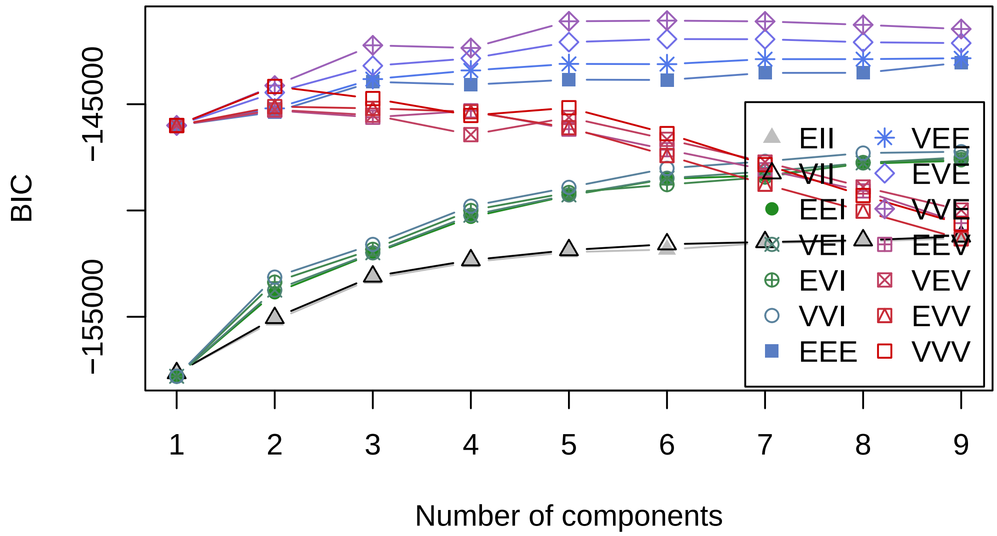



Figure 12: **Baysian Information Criteria Plot for Choosing Optimal Model**

VVE was found to be the best performing model. Promisingly, the plot peaked at four clusters. Since it is known that there actually are four categorical classes in the original data, this shows that the model is learning from the distributions of all four labels. Since we saw such overlap in classes and many misclassifications in class D previously, it would have been unsuprising to see three as the optimum number of classes,suggesting that data was better described by three categories than four.

In heirarchical clustering, we didn't see clear confirmation of four groups from the graphs using either the elbow method, gap method, or silloette method.

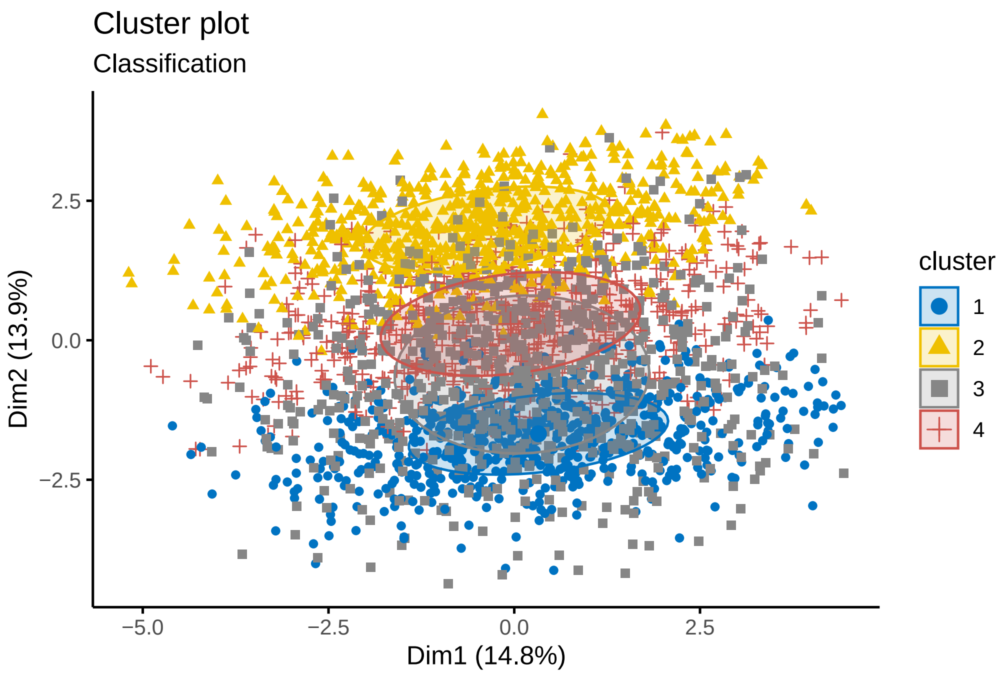
The three letters in the model name describe the shape, orientation, and orientation of the clusters that the model predicts. In this case, VVE indicates that the model is predicting clusters that are elipsoids of equal orientation, but varying volume. This makes sense based on the PCA scatterplots where the data is shown as three roughly equally size elipses one above the other, with a fourth, larger elipses overlaying them, with the major axis of all four being roughly horizontal (along the first principle component)

```
----------------------------------------------------
Gaussian finite mixture model fitted by EM algorithm
----------------------------------------------------

Mclust VVE (ellipsoidal, equal orientation) model with 4 components:

 log-likelihood    n  df       BIC     ICL
      -69141.02 2775 353 -141080.8 -141442
```

```
Clustering table:
  1   2   3   4 
700 695 677 703 
```




This looks really similar to our original PCA plot, with class A B and C seperated and class D overlapping all three.

Table 33: Gaussian Mixture Model Confusion Matrix

| | Reference | | | |
|---|---|---|---|---|
| Prediction | A | B | C | D |
| A | 641 | 0 | 0 | 62 |
| B | 0 | 662 | 0 | 38 |
| C | 0 | 0 | 677 | 18 |
| D | 61 | 32 | 60 | 524 |

Table 34: Gaussian Mixture Model Overall Statistics

| | Statistic | Value |
|---|---|---|
| Accuracy | Accuracy | 0.9023 |
| Kappa | Kappa | 0.8698 |
| AccuracyLower | AccuracyLower | 0.8907 |
| AccuracyUpper | AccuracyUpper | 0.9131 |
| AccuracyNull | AccuracyNull | 0.2656 |

| | | |
|---|---|---|
| AccuracyPValue | AccuracyPValue | 0.0000 |
| McnemarPValue | McnemarPValue | NaN |

Table 35: Gaussian Mixture Model Statistics by Class

| Statistic | Class: A | Class: B | Class: C | Class: D |
|---|---|---|---|---|
| Sensitivity | 0.9131 | 0.9539 | 0.9186 | 0.8162 |
| Specificity | 0.9701 | 0.9817 | 0.9912 | 0.9283 |
| Pos Pred Value | 0.9118 | 0.9457 | 0.9741 | 0.7740 |
| Neg Pred Value | 0.9706 | 0.9846 | 0.9712 | 0.9438 |
| Precision | 0.9118 | 0.9457 | 0.9741 | 0.7740 |
| Recall | 0.9131 | 0.9539 | 0.9186 | 0.8162 |
| F1 | 0.9125 | 0.9498 | 0.9455 | 0.7945 |
| Prevalence | 0.2530 | 0.2501 | 0.2656 | 0.2314 |
| Detection Rate | 0.2310 | 0.2386 | 0.2440 | 0.1888 |
| Detection Prevalence | 0.2533 | 0.2523 | 0.2505 | 0.2440 |
| Balanced Accuracy | 0.9416 | 0.9678 | 0.9549 | 0.8722 |

Gaussian mixed model clustering performed very well. It immediately had overall accuracy comparible with random forest, and performed notebly better than other models at seperating the fourth class successfully, with an F1 score of 0.8.

Attempts were made to further tune the model through regularization:

Table 36: Gaussian Mixture Model Performance with Different Regularization Settings

| | Model | ARI | Accuracy | F1 Score (A) | F1 Score (B) | F1 Score (C) | F1 Score (D) |
|---|---|---|---|---|---|---|---|
| Accuracy | No regularization | 0.7702 | 0.9023 | 0.9125 | 0.9498 | 0.9455 | 0.7945 |
| Accuracy1 | Shrinkage = 0.001 | 0.6016 | 0.7643 | 0.8096 | 0.8876 | 0.8795 | 0.1837 |
| Accuracy2 | Shrinkage = 0.01 | 0.5611 | 0.7571 | 0.7984 | 0.8995 | 0.8768 | NA |
| Accuracy3 | Shrinkage = 0.05 | 0.5770 | 0.7575 | 0.7995 | 0.8977 | 0.8789 | NA |
| Accuracy4 | Shrinkage = 0.1 | 0.6016 | 0.7643 | 0.8096 | 0.8876 | 0.8795 | 0.1837 |
| Accuracy5 | Shrinkage = 0.5 | 0.5640 | 0.7575 | 0.7984 | 0.8995 | 0.8783 | NA |

The best performing model did not use regularization.

The motivation for using regularization is to help the model perform better when using features that correlate together. Since regularization in fact hindered performance, another approach is to use factor analysis to reduce the dimensionality of the data. Factor analysis is suitable because it is suited to handling the groups of correlated features notable in the EDA results, and because the data is biological in origin. With biological data, there is often an underlying cause, like a gene expression, that can have many measurable implecations, like disease symptoms or physical characteristics. Because we know these mechanisms may exist in the source of our data, factor analysis is a good choice to reduce dimensions while preserving as much information as possible.

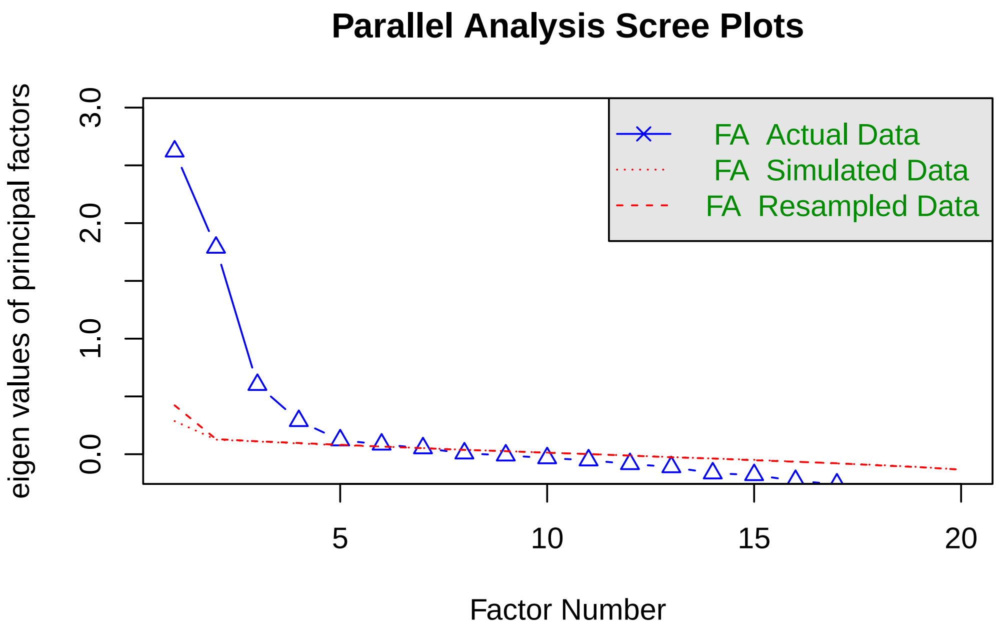



Figure 13: **Scree Plot:** *for Choosing Number of Factors for Dimensionality Reduction*

The parallel analysis results suggest that the optimum number of factors is 6. Using the maximum likelyhood method finds the number of factors where the value of the eigenvalue is above what would be expected by random chance.

The underlying values from the analysis show that the first two factors contribute the most, with a sharp drop after that. So a dimensionality reduction to two factors could be a reasonable option. We can also see that the seventh and eigth eigenvalues are not much smaller than the sixth, so swapping some of the smaller factors could also be a justifyable experiment.

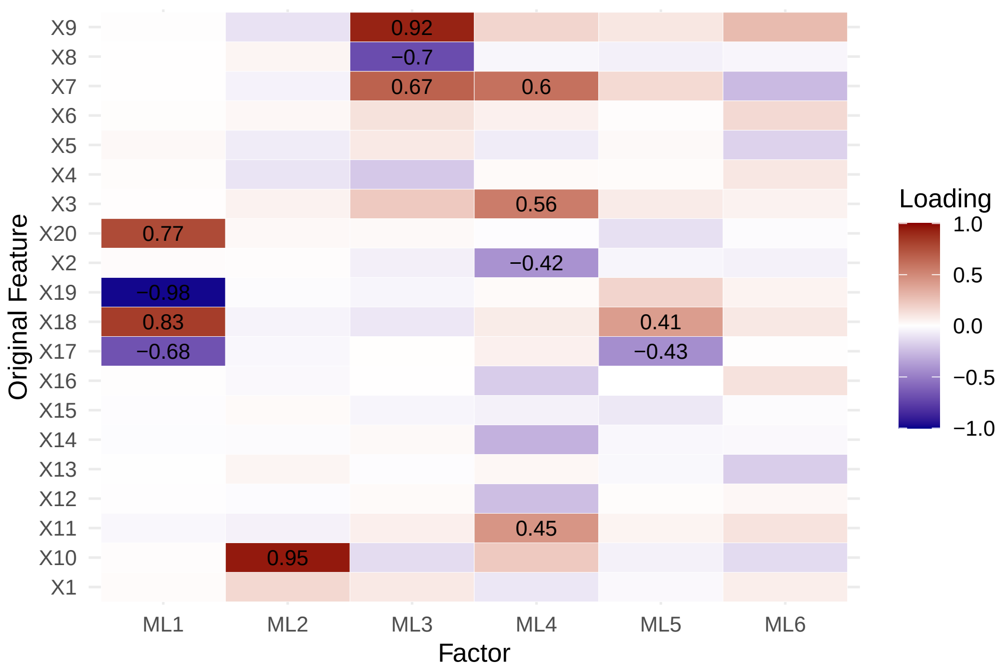



Figure 14: **Factor Analysis Loadings:** *loadings values by feature*

The loadings of features to factors shows the highly correlated features in the correlated groups are all heavily loaded to the same factor, which will succesfully decrease the multicolinearily present in the dataset.

Table 37: Gaussian Mixture Model Confusion Matrix

| | Reference | | | |
|---|---|---|---|---|
| Prediction | A | B | C | D |
| A | 650 | 4 | 1 | 77 |
| B | 10 | 669 | 0 | 37 |
| C | 2 | 0 | 728 | 41 |
| D | 40 | 21 | 8 | 487 |

Table 38: Gaussian Mixture Model Overall Statistics

| | Statistic | Value |
|---|---|---|
| Accuracy | Accuracy | 0.9023 |
| Kappa | Kappa | 0.8698 |
| AccuracyLower | AccuracyLower | 0.8907 |
| AccuracyUpper | AccuracyUpper | 0.9131 |
| AccuracyNull | AccuracyNull | 0.2656 |
| AccuracyPValue | AccuracyPValue | 0.0000 |
| McnemarPValue | McnemarPValue | NaN |

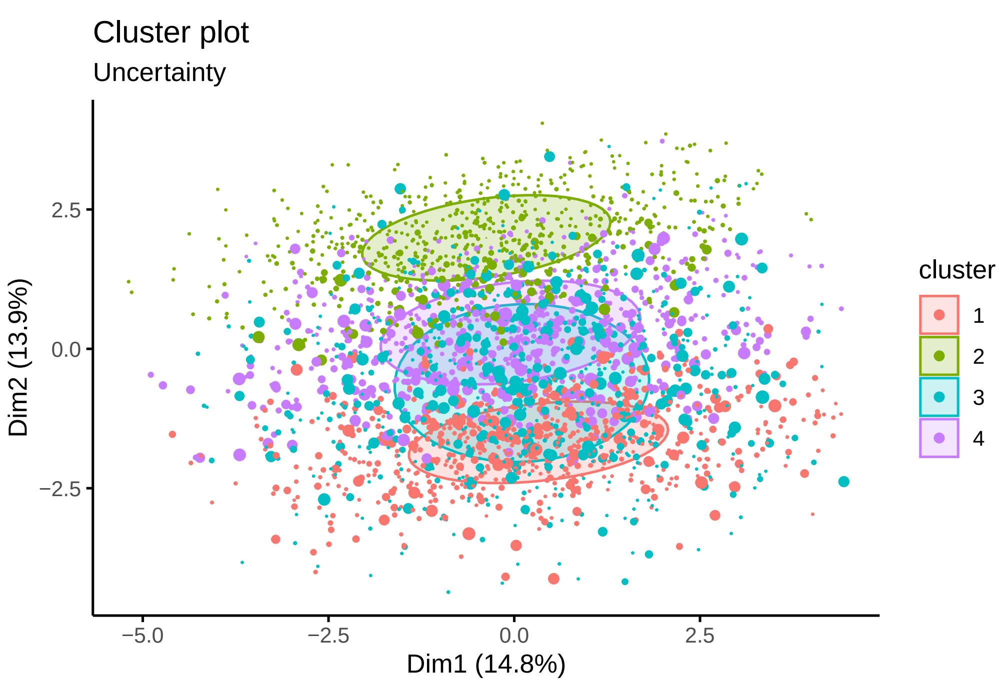
Table 39: Gaussian Mixture Model Statistics by Class

| Statistic | Class: A | Class: B | Class: C | Class: D |
|---|---|---|---|---|
| Sensitivity | 0.9259 | 0.9640 | 0.9878 | 0.7586 |
| Specificity | 0.9604 | 0.9774 | 0.9789 | 0.9677 |
| Pos Pred Value | 0.8880 | 0.9344 | 0.9442 | 0.8759 |
| Neg Pred Value | 0.9745 | 0.9879 | 0.9955 | 0.9301 |
| Precision | 0.8880 | 0.9344 | 0.9442 | 0.8759 |
| Recall | 0.9259 | 0.9640 | 0.9878 | 0.7586 |
| F1 | 0.9066 | 0.9489 | 0.9655 | 0.8130 |
| Prevalence | 0.2530 | 0.2501 | 0.2656 | 0.2314 |
| Detection Rate | 0.2342 | 0.2411 | 0.2623 | 0.1755 |
| Detection Prevalence | 0.2638 | 0.2580 | 0.2778 | 0.2004 |
| Balanced Accuracy | 0.9432 | 0.9707 | 0.9833 | 0.8631 |

Operating on the factors rather than the original features, overall accuracy is slightly higher and the balanced accuracy of cluster 3 (class D) is slightly lower. This is a trade off that requires more knowledge of the intended use of the model to make a choice between the two.

However, the fact that factor analysis leads to increased accuracy overall while decreasing the dimensionality of the data so far is an interesting finding.

#### 3.3.2.1 Model Evaluation



```
 A  B  C  D 
23 10  6 38 
```

```
# A tibble: 10 x 4
# Groups:   cluster [4]
   cluster true_label     n percentage
     <dbl> <fct>      <int>      <dbl>
 1       1 B            662      94.6 
 2       1 D             38       5.43
 3       2 C            677      97.4 
 4       2 D             18       2.59
 5       3 D            524      77.4 
 6       3 A             61       9.01
 7       3 C             60       8.86
 8       3 B             32       4.73
 9       4 A            641      91.2 
10       4 D             62       8.82
```

## 4 Discussion

the final model had good overall accuracy but caution is advised when using a ml model with this data due to the poor performance in class D - if false positives or false negatives in this class have serious implications, some models become immediately unusable.

## 5 Conclusion

## 6 References