CFAS420 Coursework: Covid-19 Classification

Lewis Tipping, 35582956

***Abstract*—This paper completes two tasks, both involving Covid CT scan data. The first task involved classifying CT scans as Covid positive/negative. XGBoost (XGB) and Logistic Regression (LR) were used. XGBoost, which is an adaptation of boosted decision trees, outperformed LR and was therefore used to predict unseen data. The second task also involved LR, but with a focus on regularisation. Models were built for both datasets successfully, scoring an AUC of 1 and 0.9. One dataset was then used to demonstrate Gaussian Mixture Models (GMM).**

I. Introduction

This project focuses on the extremely relevant project of Covid-19 detection. There are two tasks, the first is to create multiple models that can classify a CT scan as being Covid positive or Covid negative. Two models were used: logistic regression and XGBoost (XGB). The second task is similar and involves modelling more Covid data that has been passed through a neural network to provide new features. Both tasks involve dimensionality reduction. The second task finishes off with clustering the reduced data which may be beneficial for alternative classification techniques.

For every model, there will be a focus on the methodology and then the experiment/results. For the first task, the best performing model will be used to predict hidden test data.

II. Simple Image Classification

III. Dimensionality Reduction

Initially, all 598 images were added to a data frame that had 40000 variables. This included the positive and negative Covid images and the test images. The dimension reduction needed to be performed on all the data so that the final model could predict on the data accurately. If only the training data were reduced, then the test data would not work with the models as the dimensionality would be different.

Principal Component Analysis (PCA) was used as the dimensionality reduction technique. The goal of PCA is to reduce the data structure whilst keeping as much relevant information as possible [1].

The data were scaled using normalisation which stops certain variables from dominating the association between variables in the PCA. The actual process involves creating a covariance matrix of the normalised data. With a matrix of pxp dimensions (40000x40000 in this case), p eigenvalues are gathered from the matrix which allows for the computation of the eigenvectors. These eigenvectors are the linear combinations that the principal components (PC's) represent. The PCs are linear combinations of the original high dimensional data, and they are ranked by the percentage of variance they explain (PVE).

A general rule is that the reduced dataset made up of the PC's should explain at least 80% of the variance in the original dataset [2]. For this reason, when PCA was applied, the first 100 PC's were used, since they explained roughly 81% of the variance. This can be seen visually in Figure 1. The first PC explains over 15% of the variance, the second 10% and so on. Even though this new dataset is large with 100 variables, it is still significantly smaller than the original 40000 variable set, which will make the models run quicker and be less taxing computationally in the model building phase.

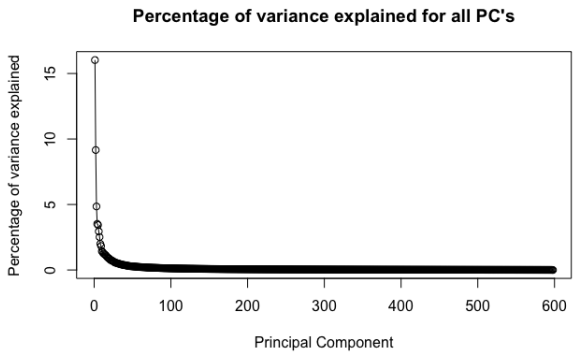

Fig. 1. Percentage of variance explained by PC's

If the original 40000 variables were labelled then it would be beneficial to investigate what each principal component represented. Loading scores show which variables are strongly represented by the new PC's. Since the variables in the original dataset are unlabelled, this was not appropriate.

The new dataset was made up of 598 observations, each of which had 100 variables. The first 500 observations were split from the final test data. A binary label was added to these 500 observations, 1 for Covid positive and 0 for Covid negative.

IV. Classification Modelling

A. Train/Test split

The data was split into training/test data. An 80/20% split was used, meaning 400 observations were used at random to train the data, and the remaining 100 observations were tested on. When models were trained, 5 fold cross-validation was used to tune the parameters. The 100 test observations were used to see how the model performed on unseen data, and it allowed the threshold to be tweaked for certain models to prioritise different success metrics e.g. sensitivity.

V. Boosted Decision trees (XGBoost)

A. Theory

The first model looked at was XGBoost (Extreme Gradient Boosting) decision trees [3]. This model is an ensemble model which uses multiple weak learners to iteratively build a model. The theory builds on normal boosted decision trees but incorporates new advancements like parallel processing and its ability to deal with sparse data. Given the size of the reduced data, it seemed appropriate to use a model that can work efficiently with a large amount of data.

For this project, the weak learners were decision trees with a maximum depth of two. These shallow trees are not very effective on their own, but when used sequentially by building on previous trees errors, they can be extremely accurate.

XGBoost can incorporate regularisation to stop overfitting, both the L1 and L2 norm can be used. These parameters were not used with this method, as regularisation is focused on more heavily in the logistic regression section.

B. Training the model

XGBoost has a plethora of hyperparameters that can be tuned. The optimal model will not be selected the first time round, it requires tweaking. To demonstrate this, the arguably two most important parameters were changed to try and optimise model performance. These parameters were the learning rate (eta) and the number of boosting iterations (nrounds). The learning rate was tested from values of 0.01 to 0.3 at intervals of 0.01. The number of boosting iterations were tested from 1 to 200. The results from 5 fold cross-validation can be seen in Figure 2. It can be seen that generally, as the number of iterations and ETA increases, the cross-validation accuracy also increases. This makes sense logically, the model has more opportunities to fine-tune its errors if the boosting iterations are high, and can quickly amend errors with a higher ETA. The optimal parameters were 129 boosting iterations with a learning rate of 0.28.

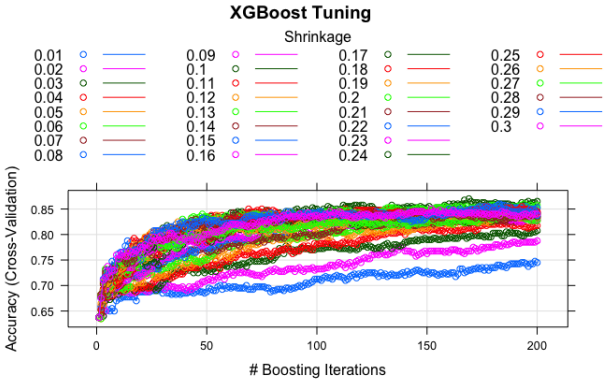

Fig. 2. XGB Tuning results from 5 fold cross-validation

When the optimal parameters were used, the average cross-validation accuracy from the model was 81.25%. These optimal parameters were then used when testing the model.

C. Testing the model

When building the model, the probabilities of the outcome being positive or negative (1 or 0) were returned. This allowed for a shifting of the threshold to cater to different performance metrics.

When the test data was predicted using the tuned model, the maximum accuracy achieved was 91%, this was achieved by setting the probability threshold to 0.497, meaning that if the predicted probability of that patient having Covid was over 0.497, they would be given a positive Covid prediction.

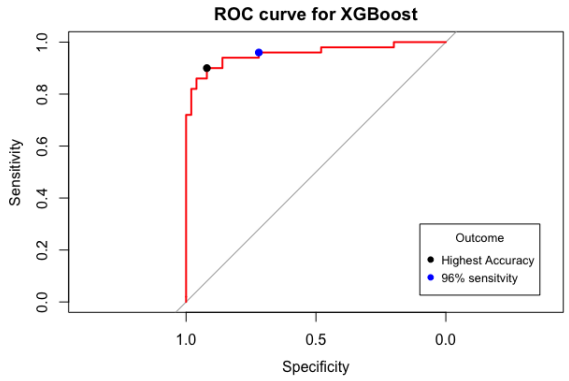

Fig. 3. ROC curve for XGBoost model

The black point in Figure 3 is where this maximum accuracy was achieved. It also achieved 90% sensitivity and 92% specificity.

It can be argued that a higher sensitivity is preferable when working with medical data. The cost of not detecting a positive Covid case is a lot higher than falsely detecting a Covid case. For this reason, the threshold can be reduced to positively detect more potential Covid cases. The obvious drawback of doing this is that when the threshold is set so low, the model will predict that the majority of people have Covid, which is not the case. The blue point in Figure 3 is where sensitivity is at 96% for the test data, meaning 96% of positive Covid cases were correctly identified. The threshold is only 0.28. With this threshold the overall accuracy was reduced to 84% and specificity was 72%. This is the trade-off demonstrated with setting the threshold so low. At each end of the ROC curve, the trade-off becomes more and more unequal, for example, increasing the sensitivity to 100% will cause the specificity to be dramatically reduced. This is because the model will use a blanket approach to give every test a positive Covid outcome, which maximises sensitivity, but means specificity is poor.

VI. Logistic Regression with regularisation

A. Theory

Logistic regression is another classification model that can be used to predict a binary outcome.

Regularisation will be demonstrated here. Three types of regularisation could be used on this data: Lasso, Ridge and Elastic Net. Ridge regression was not appropriate for this data set. From the PVE it can be seen that only the first 10 or so

variables contain over 1% of the variance explained. It would be unnecessary to force all 100 variables to be in the model through ridge regression. Elastic net performs better when there are more observations (p) than variables (n), in this case, $p \not> n$. Elastic net also performs better when the covariates are highly correlated, there is a grouping effect [4] and variables are included/excluded based upon group characteristics. Since PCA has been performed, this is not the case.

This leaves Lasso regression. Lasso regression works well when there is a lot of variables since it has the power to discard irrelevant coefficients [5] unlike Ridge regression. For this reason, it shall be used. This involves setting alpha as 1 in the R code. A worthy side note is that setting alpha as 0 would lead to ridge regression and setting alpha between 0 and 1 would lead to Elastic Net since Elastic Net is a weighted average of the two.

B. Training the model

As regularisation is being introduced, the first step in training the model is setting an appropriate lambda (λ). This lambda determines the level of shrinkage. The goal is to have a model that has the lowest error rate, but there is a trade-off that the model should be as simple as possible and not overfit the training data. A grid search was used to find the optimal lambda value, and then the one standard deviation rule was applied to find the lambda that returned the simplest model that had an error rate within one standard deviation away from the optimal lambda.

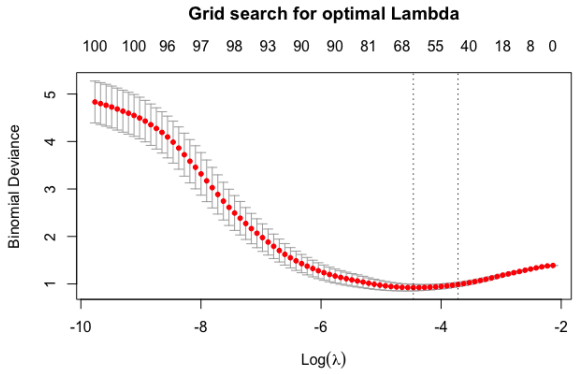

Fig. 4. Grid Search for optimal lambda

Figure 4 shows the grid search. The dotted line on the left reveals the log lambda that returns the smallest prediction error ($\lambda = 0.011$). The dotted line on the right is the largest lambda value that returns an error rate within one standard deviation of the minimum ($\lambda = 0.024$). The second lambda value will be the one used to build the model.

When the one standard deviation lambda is used in the model building process, the number of PC's included within the model is 46. 68 variables were included if the original lambda value was used. This regularisation method can reduce the number of coefficients by almost half, at the small cost of a one standard deviation increase in error.

When building the model, there are more regularisation parameters to set, the complexity parameter used was the AIC. The AIC method will tend to choose a more complex model compared to other methods i.e. BIC [6]. The AIC is more appropriate when a false negative is more damaging than a false positive, which is often the case with medical data [7].

C. Testing the model

The model was tested which allowed for a comparison to the XGBoost model. The same principles followed as before, the predictions returned are probabilities that the observation was either positive or negative (1 or 0). An ROC curve could then be created similar to the XGBoost model, as seen in Figure 5.

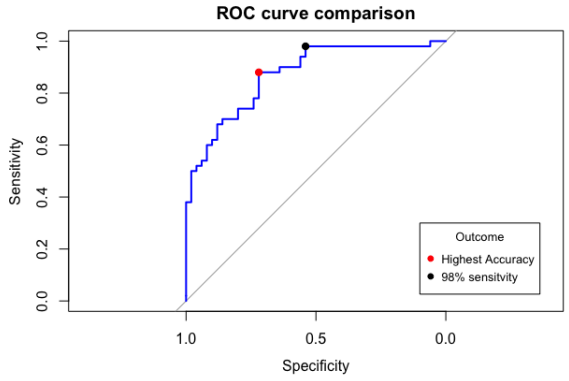

Fig. 5. ROC curve for logistic regression

The optimal point on the ROC curve is the red dot. When the threshold that leads to this point is chosen, the model was 80% accurate on the test data. The sensitivity at this point was 88%, and the specificity was 72%.

VII. Model Selection

To compare the two models, the ROC curves can be plotted against each other. Figure 6 can give a good visual comparison of the two models. A good way to judge a model is by how tightly the ROC curve hugs the upper left of the axis and thus maximising the AUC metric. The XGBoost model, in red, clearly outperforms the logistic boost model at all levels of the curve.

The XGBoost model has been extremely popular in the DS landscape due to its strong performance in terms of accuracy and computational speed. In this case, it performed better than a regularised logistic regression model. When regularisation is applied, the accuracy of the model can be limited in an attempt to stop overfitting.

XGBoost was used to predict the final test data. The threshold was set such that accuracy was maximised, this is because the trade-off for increasing sensitivity by 6% was reducing accuracy by 19%. This means a threshold of 0.497 was used.

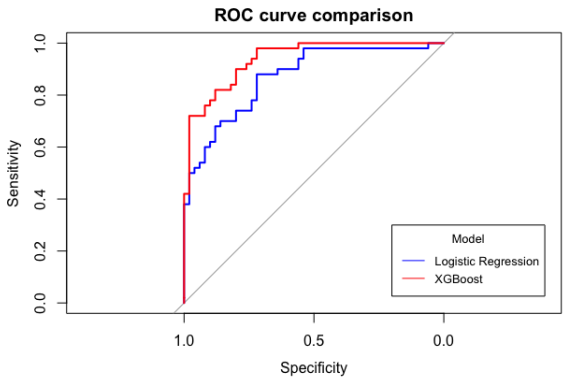

Fig. 6. ROC curves for both classification models

VIII. Modelling Neural Network Features

IX. Visualisation

For the second task, the dimensions for both datasets need to be reduced before they can be visualised. Once again PCA was performed. For both datasets, some columns are completely made up of 0 entries. This means that scaling within the PCA function is not appropriate since the function will try to scale the 0 entries which requires dividing by 0. A loop was created that will identify non 0 columns and scale them prior to PCA, whilst leaving the 0 columns as they are.

Once the data was scaled, PCA could be performed. The imagenet data was used first, Figure 7 shows the relationship between PC1 and PC2. The plot shows the 500 observations represented by the principal components. The principal components are linear combinations of the original 4096 variables. PC1 and PC2 explain the largest portions of the original data. It is hard to distinguish a Covid positive case (labelled 1) from a Covid negative case (labelled 0) which may be an issue when classification occurs.

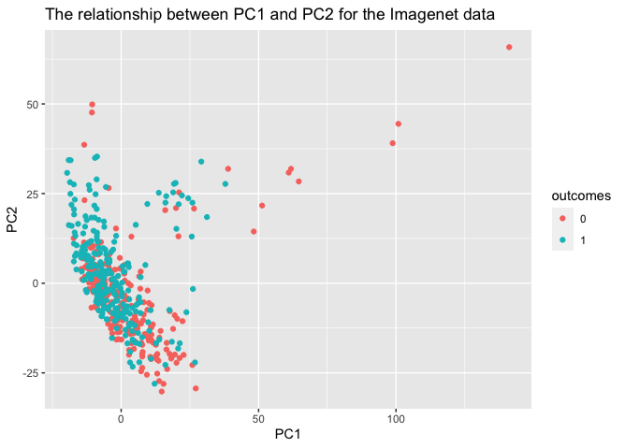

Fig. 7. PC1 plotted against PC2 for the imagenet data

The same technique was applied to the tuned data, the results (Figure 8) show a more structured separation between a positive and negative Covid case. The pattern seems to indicate that if both PC1 and PC2 are negative, the observation does not have Covid.

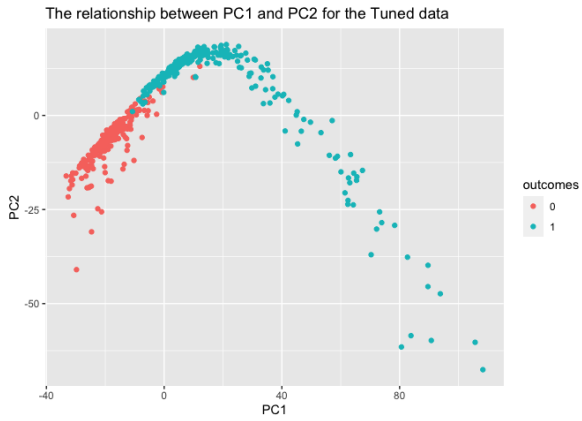

Fig. 8. PC1 plotted against PC2 for the Tuned data

The strength of PCAs performance can be determined by the portion of variance that the first few PC's represent. As mentioned previously, the goal is for the newly structured reduced data to represent 80% of the variance. Figure 9 shows how much variance the first 10 principal components explain. It is clear that the tuned data is more suited to the application of PCA, with the first 10 components explaining 48% of the variation, compared to the first 10 imagenet PCs explaining 31%. One interesting note is that it is only the first PC that is significantly different between the two datasets, which could be a byproduct of the tuning. Given that the tuned data works better with PCA and that there is a clear visual split between results, it can be expected that the tuned data will perform better under most classification models.

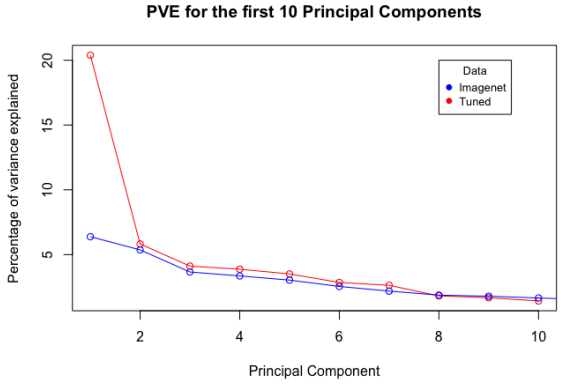

Fig. 9. Percentage of variance explained for first 10 PC's

X. Image Classification - Logistic Regression

A. Data

For this task, two datasets are being used. Both of these datasets are made up of 500 variables of 4096 observations. An outcome variable was bound on to each data frame indicating

whether that person had Covid, a value of 1 for Covid Positive and 0 for Covid Negative.

B. Regularisation

Since both datasets have 4096 covariates, a good regularisation technique is essential.

As was mentioned earlier, three types of regularisation can be considered here. Lasso regression (L1 norm), Ridge regression (L2 norm) or Elastic net can be chosen. The type of regularisation depends on the data requirements. For these data, the number of covariates needs to be reduced, otherwise, the model will be far too complex. This means Lasso regression will be more appropriate.

Lasso regression can make covariate coefficients 0, essentially removing that covariate from the model. Ridge regression, on the other hand, can only shrink the covariates, meaning that the size of the model is not reduced. Elastic net is a weighted average of the two L1 and L2 penalties.

Lasso will be used here. The first task is to find an appropriate lambda such that the model is sufficiently reduced in size, but without too much of a loss in terms of accuracy. A grid search was applied, and varying lambda values were tested for both datasets.

Focusing first on the tuned data, Figure 10 shows the grid search for the optimal lambda. The dotted line on the left was the lambda value which returned the smallest cross-validation error ($\lambda = 0.0043$). The one standard deviation rule was applied, and the dotted line on the right is where the largest lambda value lies when the error rate is one standard deviation away from the minimum ($\lambda = 0.0060$).

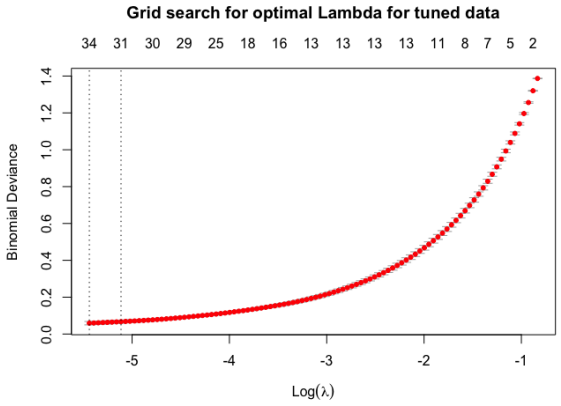

Fig. 10. Grid search for optimal lambda with tuned data

The one standard deviation lambda was chosen, this omitted 4065 of the covariates, leaving only 31 variables in the model. This is known as being a sparse model since the relevant features are spread far apart.

This was also applied to the imagenet dataset, as seen in Figure 11. The error minimising lambda was 0.009 and the chosen lambda was 0.021. Only 74 covariates were included in this model, once again showing the power of regularisation.

An interesting point is the effect of the standard deviation rule. For the tuned data, the size of the model is only shrunk by

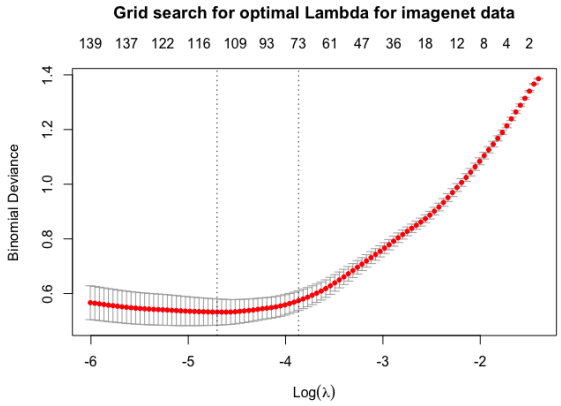

Fig. 11. Grid search for optimal lambda with imagenet data

3 covariates when the lambda is changed from the minimum error value to the one S.D. rule value. In comparison, the imagenet model has 37 covariates removed when this rule is applied. It can be seen that the actual shrinkage itself is more beneficial to the tuned data since the number of coefficients is lower, but the one standard deviation rule is more beneficial to the imagenet model, as a larger percentage of remaining variables are removed when this rule is applied.

These grid searches demonstrate the trade-off faced between bias and variance. Variance can be reduced by having more covariates, but this leads to a model that will be too specific to the training data, and would not perform well in a different scenario.

C. Logistic regression models

Now that an optimal Lambda had been chosen, the models could be built. The models were trained using 5 fold cross-validation. The parameters were set up using a tuning grid. The one S.D. lambda values were used. The models were then used to predict the Covid outcome on the relevant test data.

These predictions allowed for two ROC curves to be created. With the ROC curve. Figure 12 shows how both models performed on the test data.

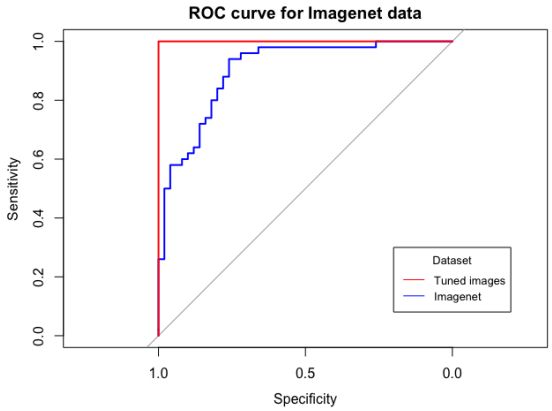

Fig. 12. The ROC curves of both models

The goal of the ROC curve is to be as close to the top left quadrant as possible, the red curve does exactly that. The tuned test dataset was predicted perfectly, the imagenet data struggled a bit more, with a highest accuracy of 85%. The thresholds for these values differed, for the tuned data, the probability that the observation had Covid had to be over 0.644 to be classed as Covid positive. For the imagenet data, this threshold was only 0.426.

Given the perfect predictions from the tuned model, it scored an AUC of 1, the imagenet model scored an AUC of 0.9. This could be because of the pre-processing phase where the tuned images were finely tuned over the last four layers of the neural network.

XI. Model Based Clustering

The tuned data were projected onto the first three principal components, meaning that a new data frame of 500 observations and 3 variables were now being used.

A. Gaussian Mixed Models

The data were clustered using a Gaussian Mixed Model (GMM). This model can test varying clusters with different specifications. There are 14 different types of cluster arrangements that can be created. The clusters distribution, shape, volume and orientation are taken into consideration [8].

The number of optimum clusters was decided by using the Bayesian Information Criterion (BIC). The aim is to maximise the BIC value.

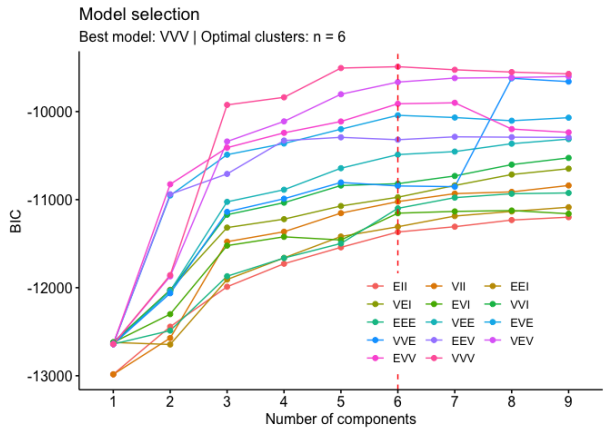

Fig. 13. Finding the optimal number of clusters using the BIC

As seen in Figure 13, the BIC is maximised when there were 6 clusters and a VVV model is used. VVV corresponds to the following parameters: an ellipsoidal distribution and varying volume, shape and orientation. This means that the clusters have the least restrictions on them out of all the possible models, which could explain why it performed best.

Each subject ID has a likelihood that it belongs to each cluster. For each cluster, the maximum cluster likelihood can be searched for in order to find the subject ID which is the most likely to be a member of the cluster. For clusters 1-6 the closest subject ID's were as follows: 247, 306, 15, 408, 499 and 256.

An alternative way to measure how similar a point is to a centroid is to use the physical distance between the centre and the given points. This differs slightly as it doesn't rely on the cluster likelihood unlike before. Instead, the subject ID that is closest to the cluster centroid can be deemed the most likely to be part of that cluster. When measuring the distances between the centroids and subject IDs multiple metrics can be used, the two most popular are the euclidean distance and Manhattan distance. The Manhattan distance metric is more appropriate for high dimensionality data [9]. Since this data set is only the first 3 PCs, the Euclidean distance metric was used. The nearest subject ID for clusters 1-6 are 182, 61, 150, 105, 227 and 144 respectively when purely measured by the Euclidean distance.

The first set of results is more appropriate as it shows how likely a subject is going to belong to that cluster, all of the subjects chosen had over a 99% likelihood of belonging to its chosen cluster. The only issue is that some of the clusters had multiple items that had a 100% chance of belonging to the cluster, at that point, the nominated subject ID is chosen from that group at random.

References

[1] H. Abdi and L. J. Williams, "Principal component analysis," *WIREs Computational Statistics*, vol. 2, no. 4, pp. 433–459, 2010. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/wics.101

[2] J. I.T., *Principal Component Analysis.*, ser. Springer Series in Statistics. Springer, 2002, vol. 2nd ed.

[3] T. Chen and C. Guestrin, "Xgboost," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug 2016. [Online]. Available: http://dx.doi.org/10.1145/2939672.2939785

[4] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

[5] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

[6] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[7] J. e. a. Dziak, "Sensitivity and specificity of information criteria," 2012. [Online]. Available: https://www.methodology.psu.edu/files/2019/03/12-119-2e90hc6.pdf

[8] L. e. a. Scrucca, "mclust 5: Clustering, classification and density estimation using gaussian finite mixture models." 2016. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5096736/

[9] C. Aggarwal, A. Hinneburg, and D. Keim, "The surprising behavior of distance metrics in high dimensional spaces," 2001.