A COMPARISON OF SUPERVISED AND UNSUPERVISED LEARNING IN THE DIAGNOSIS OF METABOLIC SYNDROME

Page count: 12 pages

Jayden Dzierbicki
MA5810

Contents

Disclaimer

I sought approval from Sourav as I have previously used this data in MA5800 capstone to ensure I avoid the issue of self-plagiarism.

Github

https://github.com/jaydendzierbicki/Machine-Learning-Metabolic-Syndrome

ABSTRACT

Machine learning in the context of healthcare is thought to grow to USD 67.4 billion industry by 2027, through the application of revolutionizing clinical decisions and making diagnosis through the use of models and artificial intelligence: with many start-ups entering and exploring this space currently through various software solutions. Too explore the possibilities of this growing industry we set out to explore and compare various machine learning models both (1) supervised models, such as Naïve Bayes and logistic regression against (2) unsupervised models such as k-means clustering and principal component analysis with k-means clustering applied as a form of dimension reduction, this was achieved by obtaining patient data which contained various numerical and categorical variables in relation to clinical and non-clinical features. Through a simple search of the literature we were unable to find any previous studies which compared and contrasted supervised learning against unsupervised in the context of metabolic syndrome, though similar studies have been applied in the context of diabetes with mixed results. A major limitation in our comparison is that by employing certain assumptions in our unsupervised learning algorithm we resulted in a loss of non-clinical binary features, thus reducing our ability to compare our models. The objective of this study was to find a model with both high accuracy and sensitivity and compare the models against one another. We found that overall accuracy of 80% was common amongst all our machine learning models, both supervised and unsupervised and suggests the predictive power of machine learning in the context of healthcare. Furthermore, through fine tuning our models through the application of assumption testing we were able to produce a model with almost 85% sensitivity, an important indicator of success used in the application of healthcare, without further compromising accuracy and specificity. It was observed that the payoff in improvement of our indicators tended to dimmish as we applied more data scientific methodologies to improve our models and meet as many assumptions as possible, with unpredictable changes such as improving and worsening some key indicators. Overall, this study found that in the application of machine learning is somewhat promising in the application of diagnosing metabolic syndrome and suggests the possibility of one day being incorporated into healthcare given the sheer amount of future investment in this space, providing benefits through the correct diagnosis of patients, and possibly removing misdiagnosis associated with errors in primary care by incorporating models into existing systems.

INTRODUCTION

Metabolic syndrome is a cluster of conditions which when present increase the risk of heart disease, stroke and diabetes through the development of insulin atherosclerosis and insulin resistance. The diagnosis of metabolic syndrome requires that a patient presents with three out of five known risk factors, which are (1) elevated waist circumference, (2) elevated triglyceride levels, (3) reduced high density lipoproteins (HDL), (4) elevated blood pressure and (5) elevated fasting blood glucose. A branch of data mining referred to as machine learning provides promise to revolutionize clinical decision and making diagnosis by making the healthcare system smarter. In recent times the use of machine learning and artificial intelligence has been projected to grow by USD 67.4 billion by 2027, a growth of 46% from today, with many companies now developing software solutions for various healthcare applications in predicting and diagnosing disease status of patients ("Artificial Intelligence in Healthcare Market worth $67.4 billion by 2027 - Exclusive Report by MarketsandMarkets™", 2021). This paper will aim to utilise various clinical parameters such as BMI and certain blood markers which have been associated with metabolic syndrome and used in previous machine learning models such as Naïve Bayes (Yu et al., 2021), in addition will also aim to find a dataset which contains ethnicity and socioeconomic statues such as income (Moore, Chaudhary and Akinyemiju, 2017) as non-clinical features when building our machine learning models. The goal of this paper will be to build both supervised and unsupervised machine learning models as they relate to metabolic syndrome and compare and contrast the (1) accuracy, (2) specificity (3) sensitivity and (4) area under the curve if applicable in our various models as possible indicators for success, and comment on the possibility and limitations of machine learning as it relates to diagnosis and comment on the ability of removing misdiagnosis in healthcare associated with primary care by physicians (Singh, Schiff, Graber, Onakpoya & Thompson, 2016).

DATA

Patients' data was sourced externally online from https://data.world/ and redownloaded on the 12th April 2022 into a specified folder and later imported into R studio for data processing. This was achieved by searching on key terms such as 'metabolic syndrome' and finding a data set which comprised of both clinical and non-clinical features. The patient's data was supplied by the user Robert Hoyt, MD (Hoyt, 2019). The data was collected and published to https://data.world/ through the use of SQL query from the Centers for Disease Control and Prevention National Centre for Health Statistics, though no further information was provided by Robert Hoyt, MD about how this was achieved or any limitations and issues.

Once the data was loaded into R studio we ensured that all white space was converted to NA values and then conducted an analysis on missing observations. We observed a total of 436 missing observations, utisling sapply() we observe missing observations in (1) maritial (208 missing), (2) income (117 missing), (3) waist circumfrance (85 missing) and (4) BMI (26 missing).

At this stage we decided to impute all our missing variables on the whole data frame for simplicity, whilst not recommended for supervised learning it allows us to produce and compare models quickly as our goal here is not to produce a definitive model but rather compare and contrast. It could be suggested that in future studies we split our data into a data frame for supervised learning, and unsupervised learning. For the supervised learning data frame we would further split it into test & train and then complete the imputation on each split to prevent data from test & train influencing one another. At this stage we would also complete an imputation on all the data for the unsupervised data frame as this uses the whole data set in building our model and are not concerned about the leakage.

Categorical Imputation: It was deemed inappropriate to attempt to predict ones marital statues, and instead decided to replace all NA values with 'unknown' utilising a base R function.

Numerical Imputation: We first created a new data frame (impute_df) and then utilised the caret package we called on the preProcess() function after creating numerous dummy variables via the dummyVars() function for our categorical features, this allows us to impute our missing variables for (1) Income, (2) waist circumference and (3) BMI through using the BagImpute method. Once we had imputed our variables we then utilised base R to replace the NA values of impute_df to our original data frame.

After completing the imputation, we then produced various boxplots looking at the relationship between each numeric feature and the status of metabolic syndrome. It was determined visually that we could distinguish some type of relationship in and difference in the boxplots between each feature.

Though for the feature urAlbCr as seen in figure 1 we could not distinguish any clear relationship, this resulted in a log transformation which produced more meaningful results as seen in appendix A, and was used going forward.

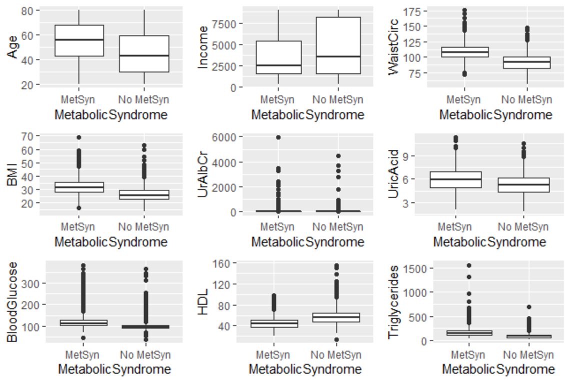


Figure 1: Multiple boxplots produces to explore relationship between numerical variables and metabolic syndrome. Visually we observe many numerical features are different depending on metabolic syndrome status. We observe that UrAlbCr provides little to no information and a log transformation was completed, the relationship after the transformation was much more obvious and can be found in Appendix A.

Furthermore, we explored our categorical variables via multiple bar charts and observe two issues, (1) we observe for race that we have a small count of observations associated with 'other' and with (2) marital statues we observe as from above we have a small number of unknown observations as seen in figure 2. Since we are unsure what 'other' stands for in the Race section we elected to remove it via the filter() function, we also elected to remove 'unknown' via the filter() function for marital status as previous studies only had defined marital status. This resulted in the removal of 263 unique rows, reducing our data frame down to 2138 unique observations for both our supervised and unsupervised learning methods. It should also be noticed that our response variable for supervised learning is not balanced as seen in figure 2, and we would benefit from applying a data scientific method which can accommodate this unbalance nature, though this is explored at a later stage.

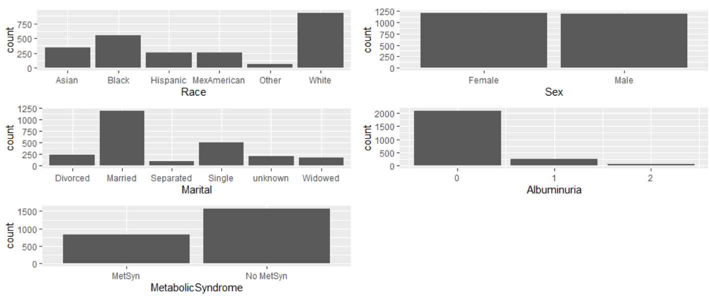


Figure 2: Multiple bar charts roduced looking at the relationship and count between metabolic syndrom and categorical varibles in the data set. Bar charts suggest that metabolic syndrome is unbalanced in nature.

Utilising the mutate() function we performed a simple coercion on all the variables as summarised in table 1 to ensure all the variables were of the correct data type to prevent the risk of upstream errors in our proposed data models. In addition we briefly describe the variable description in the model.

Table 2: Variable types and coercion applied during data pre-processing, along with description of varible.

Variable Name	**Datatype/R Coercion**	**Description**

Seqn	Integer	ID varible
Age	Integer	
Sex	Factor/*as.factor()*	Gender – 2 levels
Marital	Factor/*as.factor()*	Marital status - 6 levels
Income	Integer	Income – No mention of time period
Race	Factor/*as.factor()*	Ethnicity – 6 levels
WaistCirc	Numeric	Waist circumference – (cm)
BMI	Numeric	BMI = kg/m^2
Albuminuria	Integer	Indicator for kidney disease
UrAlbCr	Numeric	
UricAcid	Numeric	Waste product associated with metabolic activity – higher levels associated with metabolic syndrome
BloodGlucose	Integer	Level of blood glucose – indicator of diabetes
HDL	Integer	High density lipoprotein – low levels associated with metabolic syndrome
Triglycerides (Tri)	Integer	"Bad" cholesterol – Elevated levels associated with metabolic syndrome
MetabolicSyndrome	Factor/*as.factor()*	Metabolic diagnosis – 2 levels

At this stage our data frame was split into two different data frames, (1) a data frame for supervised learning which contained both numerical and categorical variables, containing 2138 observations and 14 features as seen in table 1, and (2) a data frame for unsupervised learning which only contained continuous numerical variables, containing 2138 observations and 10 features, essentially omitting the data from table 1 which is coerced as.factor(). Further processing is described in the method section as it relates the specific tests, such as removing patient ID, and removing metabolic syndrome diagnosis for unsupervised learning.

METHOD

A number different machine learning algorithms were conducted, with supervised being conducted on a training data set and unsupervised method being conducted on the complete data set as described above. The algorithms employed are (1) Naïve Bayes and (2) logistic regression which are both a form of supervised learning which have been reported and used in previous studies looking at metabolic syndrome (Yu et al., 2021), in addition we also did (3) k-means clustering and (4) k means clustering with PCA applied. All three algorithms were conducted with R and RStudio(Version 1.4.1106) on a x86_64-w64-mingw32 platform. Additional data processing steps are also summarised in this section, and each step ensured we utilised the set.seed() function for reproducibility via selecting the seed as 2343. An important assumption made at this stage is that the diagnosis of patients with metabolic syndrome in the data set is correct, and no errors are made in the source data, that is no patient is assumed to have been misdiagnosed.

Supervised learning - Method

Naïve Bayes – Assumption testing method

Naïve Bayes (NB) is a form of supervised learning which allows us to utilise both our categorical and numerical variables when building and designing a model, allowing us to incorporate both clinical and non-clinical risk factors into our model. NB classification is based on the Bayes theorem with the main assumption of independence between features. NB assumes that there is conditional independence amongst predictors and the response variable is categorical, we know from figure 2 that our response variable is categorical with two levels. For the purpose of testing independence, we assume a correlation coefficient whose magnitude are less then +/-0.39 as 'independent' for the purpose of assumption testing as we consider this "weak" association in the context of health application (Schober, Boer & Schwarte, 2018). From the correlation matrix in appendix A we observe the following correlations exceeding this standard as summarised in table 1. It was decided that since the diagnosis of metabolic syndrome is a function of HDL and triglycerides that we retain them in our model, and that they are also borderline "moderate" association (Schober, Boer & Schwarte, 2018). In addition, we created a new data frame and removed BMI for comparison, we elected to remove BMI as a diagnostic feature of metabolic syndrome requires an enlarged waist circumference.

Table 2: We flagged correlations with an absolute value equal to or above 0.39 as possible violations of the independence assumption. It was decided to retain both HDL, Triglycerides, and waist circumference as they are used in the diagnostic features of metabolic syndrome, we elected to drop BMI due to the high correlation and produce a new data frame to compare the effects of removing the feature.

Feature 1	Feature 2	Correlation
HDL	Triglycerides	-0.39
BMI	Waist Circumference	0.91

Naïve Bayes – Preliminary model development method

We then commenced building our model and called on the caret package, which allowed us to use the createDataPartition() function to split our data into a train and testing set at 80%, we then reviewed the distribution of various features by utilising the hist() function. We developed two different Naïve Bayes models at this stage by calling the naivebayes package and naivebayes() function, with the main different by setting usekernal as false and true respectively in our naïve bayes model to produce both parametric and non-parametric naïve bayes models, we then removed BMI feature on our test and training data frame and set usekernal as true in our third naïve bayes model. We utilised the caret package to produce multiple confusion matrix via the confusionMatrix() function and called on the pROC package to calculate the area under the curve (AUC) via the roc() function.

Naïve Bayes – Final model development method

We developed a fourth model utilising the same training data used in model 3 as it performed best as defined by AUC found in our preliminary model development. This time we applied synthetic minority oversampling technique (SMOTE) to the NB model to account for the unbalanced data, as it has been used in similar models which have looked at NB as it relates to metabolic syndrome status (Kim et al., 2022). We achieved this via the train() and trainControl() function and setting sampling to "smote", we then utilised the confusionMatrix() and roc() function as above to obtain our key indicators.

Logistic Regression – Preliminary model development method

Logistic regression (LR) is another form of supervised learning which allows us to utilise both our numerical and categorical variables through the application of dummy variables. This model allows us to obtain the odds ratio in the presence of multiple explanatory features. In preparation for our model development, we removed metabolic syndrome status from the data frame and then utilised dummyvars() function to create dummy variables for all our categorical variables, once completed we utilised cbind() to join on metabolic syndrome diagnosis to the data frame with numerical and dummy variables. We then converted all dummy variables to factors utilising the mutate() function, including metabolic syndrome diagnosis and confirmed this had been actioned correctly by calling on str() function. Furthermore, we called on the createDataPartition() function to split our new data frame into a train and testing set at 80%.

Our first logistic model utilised the glm() function by setting family binomial link as "logit", we set the threshold for diagnosis at 0.5 and called on the confusionMatrix() function and roc() function. Our next two models utilised two different cross validation methods for comparison, this was achieved by calling on the trainControl() function and setting method to "LOOCV" for leave one out, and the other method to "repeatedcv" and setting the number to 10 with 5 repeats, both these parameters where then placed withing the train() function calling on the "glm" method and "binomial" family. We then called on the predict() function to predict our response against our test set and used confusionMatrix() and roc() function to obtain our indicators for comparison.

Logistic Regression – Final model development method

We developed a forth model utilising the same training data set used in the previous models, and selected a cross validation method with the best indicators. Though this time also applied SMOTE to the logistic model to account for the unbalanced nature. The main difference with the above models is that in the trainControl() function we included a new parameter of sampling and set that to "smote" whilst setting method to "LOOCV", all other steps are identical, such as calling on the train(), predict(), confusionMatrix() and roc() function to obtain our key indicators for comparison.

Logistic Regression – Assumption testing method & review of coefficients

We tested the following assumptions against our final model, (1) no multicollinearity among our predictors, (2) no extreme outliers and (3) a linear relationship between predictors and the logit of the response.

When testing for multicollinearity via car package and the VIF() function we obtained an error suggesting of perfect multicollinearity as seen in appendix A. Too overcome this we reviewed the summary() of the logistic model and observed 3 coefficients with NA, indicating perfect multicollinearity ("R - dealing with new aliased coefficient ("NA" coefficient) in categorical variables for VIF", 2020). We created a new test and train subset by removing the following dummy variables (1) Sex.Male, (2) Marital.Widowed and (3) Race.White from both test and train data frame previously obtained, we then created a fifth logistic model following the exact same steps in the final model development above, this time using a the new and reduced test and train data set. We will use this new model going forward in the assumption testing. We then called on the vif() function to test our new model and reviewed the coefficients via summary() function.

We tested for extreme outliers by calling on the broom package and the augment() function to produce the standard residuals, we then filtered our rows via the filter() function and selecting for only when the absolute value of standard residual was in excess of 3.

We tested for the assumption of linearity, though because SMOTE was applied it synthetically created new rows as a product, this resulted in us having to use logistic model 2 when plotting for linearity as we encountered issues due to difference in number of rows in our final model. We start of by predicting the probability of our model response and predict our class by selecting the generic 50% threshold as a cut-off, this is achieved via the predict() function and ifelse() function respectively, using logistic model 2. We then create a new data frame and select for numeric features only and create a new variable and bind the logit, we then created multiple scatter plots utilising ggplot2.

Unsupervised learning Method

The k-means clustering methods is a form of unsupervised learning which takes *n* observations and an integer defined as *k.* The output produced of n observations into k sets ensures that each observation belongs to one cluster with the nearest means. For the purpose of selecting a value *k* we elected the value of 2 based on domain knowledge of the data set and elect to use the whole data set, though we remove the outcome variable at this stage via the select(-) function. A limitation of k-means clustering is that we require all variables to be continuous as a major assumption (K-Means Cluster Analysis | Columbia Public Health, 2022), for the purpose of this model we call on the unsupervised learning data frame previously developed to ensure this assumption is satisfied.

k-means – Model development method

We developed two new data frames, one retaining metabolic syndrome diagnosis and another removing metabolic syndrome diagnosis to ensure our data frame is unlabelled in nature. As our continuous variables are recorded in various different scales we applied the scale() function to our data frame as recommended for k-means clustering with different scales (James et al, 2021). We then called on the kmeans() function with centres set to 2 and nstart set to 25. We then called on cbind() to our unscaled data frame with predicted clusters and applied lapply() to obtain the mean of each continues varible by predicted cluster type. Furthermore we called on cbind() for our cluster prediction and known diagnosis together and called on the table() function, this allowed us to produce an accuracy measure by summing the diagonal of the contingency table with the total sum of the contingency table.

k-means with PCA – Model development method

A common technique employed to reduce noise via dimension reduction is to apply principal component analysis (PCA) prior to the development of a k-means model (Ding & He, 2004). In an attempt to improve our models accuracy we called on the prcomp() function and applied it to the previously scaled data frame as obtained above. We employed the elbow method by calling on fviz_eig() which allowed us to select the number of principal components (PC) used in our k-means model. Once we knew the number of PC to use in our study we selected the columns and inserted our new dimension reduced data frame into the kmeans() function and followed the steps outlined above to produce our contingency table to calculate our key indicators.

RESULTS

Naïve Bayes - Results

We observe in table 3 four different NB models produced, Naïve Bayes model 4 exceeds all other models in accuracy, specificity and area under the curve (AUC), whilst it comes in second place for positive predictive value. It should be noted that Naïve Bayes model 2 performed the best in both sensitivity and negative predictive value, whilst Naïve Bayes 4 came in third place for sensitivity, though this difference was only 0.0269 and is larger increase to our initial Naïve Bayes 1 model. It should be noted that accuracy is a poor indicator for our models except Naïve Bayes 4 as we applied SMOTE to that model, and this is due to our other models having bias present due to the unbalanced nature. We overall see an increase in performance based on our indicators in table 3 through the use of applying a non-parametric test to all our models, and as we further refine our model we obtain mixed trade-offs, such as increasing AUC whilst at the same time decreasing sensitivity in Naïve Bayes 4 relative to Naïve Bayes 3.

Table 3: A summary of performance for various Naïve Bayes models, we tend to observe that as we refine our model by adjusting for the distribution of the underlaying data and ensuring assumptions are withheld that we see an overall improvement in our model for all indicators except positive predictive value.

	Naïve Bayes 1	**Naïve Bayes 2**	**Naïve Bayes 3**	**Naïve Bayes 4**

Note	Use kernel = False	Use kernel = True	Removal of BMI and inclusion of use kernel = True	Removal of BMI, use kernel = True & use of SMOTE sampling
Accuracy	0.7635	0.8080	0.8126	0.8150
Sensitivity	0.5772	0.8725	0.8523	0.8456
Specificity	0.8633	0.7734	0.7914	0.7986
Pos Pred Value	0.6935	0.6736	0.6865	0.6923
Neg Pred Value	0.7921	0.9188	0.9091	0.9061
AUC	0.7202	0.8229	0.8219	0.8221

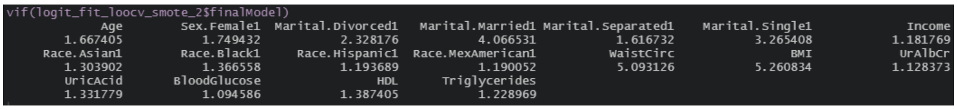
Logistic Regression – Results

We observe in table 4 five logistic models developed during the model development for logistic regression, it should be noted that without a form of cross validation as seen in logistic 1 that we have a model which performs poorly across all our indicators, only producing an AUC of 0.10791. We observe that when we perform a cross validation method such as leave one out or a 10 fold cross validation that our model improves drastically (700.76% increase in AUC), without any difference between indictors monitored in table 3 between each cross validation method. As with the Naïve Bayes results in table 3, we attempted to improve our model due to the unbalanced nature by applying SMOTE to our model, this seen a 4.6% increase in AUC and 11.9% increase in sensitivity in our logistic model 4 relative to model 2 & 3 as seen in table 4.

Table 4: A summary of performance for various logistic models, we observe that when we fail to performe a cross validation method as seen in logistic 1 that our model perfomes very poorly, whilst both cross validation methods results in the same outcome in our key indicators monitored here.

	Logistic 1	**Logistic 2**	**Logistic 3**	**Logistic 4**	**Logistic 5**
Note	Generic	Cross Validation: Leave one out	Cross Validation: 10 fold	SMOTE & Cross Validation: Leave one out	Same as logistic 5, though removed multicollinearity variables.
Accuracy	0.1616	0.8384	0.8384	0.8244	0.8173
Sensitivity	0.26174	0.7383	0.7383	0.8389	0.8389
Specificity	0.10791	0.8921	0.8921	0.8165	0.8058
Pos Pred Value	0.13589	0.7857	0.7857	0.7102	0.6983
Neg Pred Value	0.21429	0.8641	0.8641	0.9044	0.9032
AUC	0.10791	0.8152	0.8152	0.8277	0.8223

Unlike Naïve Bayes, we test our model assumptions after developing the model, the first test conducted was looking for multicollinearity, at this stage we encountered an error in logistic 4 when calling on the VIF function, this resulted in us developing a fifth model and was a result of the dummy variable trap in regression modelling (Karabiber, 2022). We observe in figure 3 that our VIF values only exceed 5 for BMI and waist circumference, whilst all other variables where below 5. In addition, we observe logistic 5 performs slightly worse relative to logistic 4, with a reduction of 0.13% decrease in the AUC, though sensitivity remained unchanged despite accounting for the dummy variable trap.



```
vif(logit_fit_loocv_smote_2$finalModel)
          Age      Sex.Female1  Marital.Divorced1   Marital.Married1 Marital.Separated1     Marital.Single1            Income
     1.667405         1.749432           2.328176           4.066531           1.616732           3.265408          1.181769
   Race.Asian1      Race.Black1     Race.Hispanic1  Race.MexAmerican1          WaistCirc                BMI           UrAlbCr
     1.303902         1.366558           1.193689           1.190052           5.093126           5.260834          1.128373
     UricAcid     BloodGlucose                HDL       Triglycerides
     1.331779         1.094586           1.387405           1.228969
```

Figure 3: We test for multicollinearity by calling on the VIF() function, we define this as a VIF value in excess of 5. We observe that BMI and Waist Circumference are both in excess of 5, whilst all other variables are less than 5.

Furthermore, we tested both the linearity and extreme outliers assumption. As seen in figure 4 we have violations in both. We observe in the left graph that variables such as age, Uric Acid and Income follow a non-linear relationship when looking at predicted value against the logit of the outcome. Furthermore, we also observe around 7 extreme outliers in the right graph due to points exceeding |3| standard residuals. This suggests that both linearity and extreme outliers is also in violation in addition to VIF.

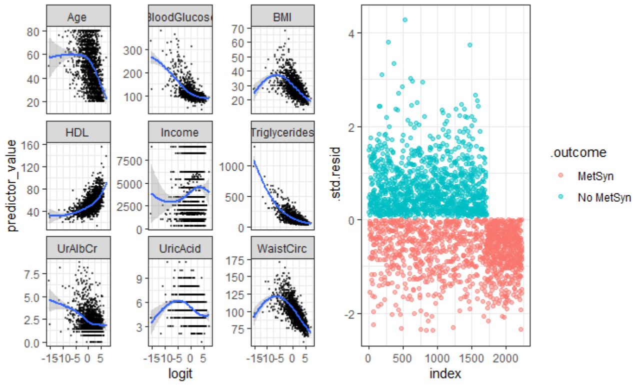


Figure 4: We test the assumption of linearity (right) and extreme outliers (left) in our logit model. For linearity we look at our continuous predictor variables and the logit of the outcome, we observe that some of our continuous features such as age, Income and UrlAcid violate this assumption visually (right). In addition, we tested for extreme outliers as defined as in excess of |standard residual| of a value of |3|, we observe that we have violations of this assumption (left).

K-Clustering & PCA – Results

Through producing various contingency tables looking at predicted cluster against known diagnosis we were able to define accuracy, sensitivity and specificity by using the equations found in Appendix A. We observe in table 5 two k-mean clustering algorithm models, both with close to 80% accuracy. It should be noted that the application of PCA lead to a slight improvement of both accuracy (0.058% increase) and specificity (0.997% increase) in our model, though the application of PCA lead to a reduction in sensitivity (16.25% reduction).

Table 5: A summary of performance of unsupervised learning technique by applying a k-means clustering algorithm with k set to 2 based on domain knowledge. Two models developed, one employing k-means clustering only and the other employing PCA prior to performing k-means clustering, a common dimension reduction technique. Through applying PCA we improved our models accuracy by an increase in 0.0004.

	K-Clustering 1	**K-Clustering 2**
Note	K = 2	K = 2, PCA applied beforehand
Accuracy	0.7946679	0.795136
Sensitivity	0.8013	0.6770
Specificity	0.7911	0.7990

Furthermore, we see in table 6 the mean values of each feature as it relates to a predicted class via k-means clustering, we generally observe a difference between each cluster and the corresponding mean value. Through looking at table 6 we observe that cluster 1 tends to be younger, higher income, healthier weight (lower waist circumfrance and lower BMI), lower UrAlbCr, lower Uric Acid, lower blood glucose, higher HDL and lower triglycerides compared to cluster 2.

Table 6 : A summary of mean values of each feature relative to the predicted cluster applied via the k-means clustering, both models produced the same mean values.

Predicted Cluster	Age	Income	WaistCir	BMI	UrAlbCr	UricAcid	BloodGlucose	HDL	Trig
1	45.6	4517.3	88.6	24.7	1.9	4.5	97.7	59.4	97
2	55.9	3582.6	111.2	33.1	2.5	5.8	123.5	45.5	169

DISSCUSION

Naïve Bayes - Discussion

We observed that when we built our Naïve Bayes model that many features followed a non-normal distribution, by accounting for that via the usekernal parameter within the naivebayes() function we were able to improve our model as seen in table 3, this resulted in an increase in all our monitored indicators except positive predicted value. It is also crucial to test assumptions as according to the assumption of independence our Naïve Bayes model 1 and Naïve Bayes model 2 was in violation of this assumption initially, prompting us to investigate. In an attempt to improve our NB models performance and overcome the violation of independence, we elected to remove BMI going forward, as it was heavily correlated with waist circumference. This resulted in some improvements in our model as seen in table 3, especially when used in conjunction with a non-parametric model, it seen an improvement in sensitivity especially. For designing a good model, sensitivity and specificity are key indicators which are used in medicine and health, with sensitivity indicating that a person with a diseases or condition has been positively diagnosed correctly. An issue with low sensitivity is delayed treatment and medical intervention whilst low specificity can result in patient stress.

A major limitation in all the Naïve Bayes model produced which did not apply the SMOTE technique is the unbalanced nature of metabolic syndrome diagnosis, as such accuracy becomes a poor measure of evaluation for our model, this is also true for the other supervised learning methods, this can be extended to our logistic model discussion as well. Previous studies have found that when applying SMOTE to a NB models which accounts for an unbalanced data set that when looking at predicting for diabetes that sensitivity increased, whilst accuracy and specificity decreased (Ramezankhani et al., 2014), though we found accuracy, specificity and AUC increased in our study as seen in table 3. In another study when SMOTE was applied to a similar unbalanced data set looking at metabolic syndrome it only resulted in a 0.091 increase in AUC value (Kim et al., 2022), which is consistent with our small increase (+0.0002) for AUC, this suggests that as we attempt to refine and improve our models that the payoff in improvement marginally increases, as seen in table 3 when comparing the increase in AUC between each model. It also suggests that other key indicators can improve and worsen in an unpredictable way based on these findings which is consistent with table 3 key indicators, this highlights a limitation in the development of a model in the application of healthcare.

Logistic Regression – Discussion

Whilst the logistic model performed better relative to Naïve Bayes as it had a relative better AUC value, all the logistic models produced violated the three assumptions which we reviewed. We could remove multicollinearity by removing BMI in a sixth model as we did with Naïve Bayes, though we would still have the issue of extreme outliers and non-linearity, with possible dimished returns as seen with Naïve Bayes model as discussed previously. We could further overcome this by possibly removing the seven extreme outliers from the data set, and possibly consider some transformation of our variables to attempt to get them to display a linear relationship. Though for the purpose of this study we did not explore this and would benefit from exploring it at a later date to see if we can further improve our models predictive power, accuracy and sensitivity. Likewise with Naïve Bayes, we are unable to comment on the accuracy of logistic model 1 through to 3 due to the unbalanced nature of the data set. It should be noted that when applying SMOTE we encountered errors with R when testing our assumption due to oversampling technique applied and resulted in us modelling this assumption on logistic model 2, we assume that the violation observed here will also apply to our models featuring SMOTE. Overall, for the application of our study we would not recommend a logistic model without further processing such as accounting for extreme outliers and the non-linear relationships. We could also attempt to improve our model by reducing dimensionality by hypothesis testing to ensure our predictors are not independent with response, though this was not explored here but is suggested as in appendix A we observe some non-statistically significant coefficients. Whilst the model lacked all assumptions, we do observe in table 4 promising results which suggests the application of machine learning in healthcare given the high sensitivity, though we must consider these violations.

K-Clustering & PCA – Discussion

A major distinction which is made here is that unlike our supervised learning model, we had a reduction in the number of features applied to our unsupervised learning method due to the assumption of requiring continuous variables for our analysis and application to the model, this limits our ability to compare our supervised learning algorithm to our unsupervised learning algorithm. Whilst PCA is employed as a method to reduce noise through dimension reduction, it lead to a reduction in our models sensitivity which is an important indicator in health and diagnosis, suggesting given the current available features we did not obtain any benefit from PCA, though it did allow us to visualise all features in 2-dimnesion as seen in appendix A which demonstrates some clear clustering. This clustering seen in Appendix A further supports the theory of the application of machine learning in the context of diagnosis.

Through analysing the mean value of each feature relative to their predicted cluster in table 6 we are able to confirm through reviewing the literature that those with metabolic syndrome tend to display characteristics with those in cluster 2, both the clinical and non-clinical features used in our model, this suggests that those in cluster 1 are those without a metabolic syndrome diagnosis, and those in cluster 2 are those with metabolic syndrome. We applied this thought when developing our key indictors such as sensitivity and specificity in our models as seen in table 5. Furthermore, natural clusters forming based on the literature and supporting the theory of machine learning in the application of healthcare.

CONCLUSION

The purpose of this paper was to compare the application of machine learning models to the diagnosis of metabolic syndrome, this was achieved by developing various supervised and unsupervised models and looking at key indicators such as (1) accuracy, (2) sensitivity, (3) specificity and (4) AUC if applicable to the model, and in addition highlight any limitations and issues faced when developing models. One major limitation which should be highlighted is the reduction and removal of features from the unsupervised learning model to ensure satisfying assumptions, meaning our unsupervised learning method and supervised learning method was built and developed based on different features, making the comparison between the two limited. This further highlight that when developing a machine learning model that it is important to test assumptions and consider many variations of the same model as part of the refinements process. It is thought and proposed here that when refining a model, the payoff in improvement will diminish, and can even impact the model in a negative way evident in the literature previously sighted. We observed that when trying to increase assumption uptake that our models did not always lead to improvements, or the improvement was dimished and impacted another measure negatively. When comparing which supervised method to elect, we would suggest a model with high sensitivity, accuracy and AUC, furthermore we would want a model which satisfies the most assumptions possible and a model which attempts to minimise bias. From table 2, table 3 and the discussion above we know that the logistic models were embed with violations, even though logistic 4 had the best AUC value as seen in table 3 (AUC 0.8277), for this reason of multiple violations we would not recommend employing a logistic regression for predicting diagnosis without further processing. Instead, we know From table 3 our Naïve Bayes model 2 had the highest sensitivity of 0.87, though an issue with this model was it was not adjusted for the unbalanced nature of the data set, thus incorporating bias into our model; for this reason and the violation of multicollinearity we would not recommend this model, instead recommend Naïve Bayes 4 which satisfies all assumptions, and accounts for the unbalanced nature of the data set.

Whilst PCA is a tool employed as a form of dimension reduction prior to commencing k-means clustering, in the aspect of our models it resulted in a 16.25% reduction in accuracy relative to our generic k-means clustering model. Despite the accuracy and specificity increasing when applying PCA, our key indicator is heavily focused on maximising specificity, and given the large percentage reduction we would not recommend this model for that reason. Furthermore, we were able to review the output of the mean value of each predictor for each class, and as expected there was a clear distinction between the mean values between each group which is consistent with the literature.

In summary, it is evident that our supervised learning algorithm was superior, with a recommendation of employing the Naïve Bayes model relative to logistic and the unsupervised learning. A major benefit of the supervised learning is we were able to incorporate both clinical and non-clinical features, as well as allowing for categorical features which are also thought to be related to metabolic syndrome such as economic status, race and even marital status. This paper overall found that machine learning in the application of healthcare yielded promising preliminary results given that accuracy tended to exceed 80% and sensitivity was also consistently around 80-85% in most models produced; and would warrant further investigation about developing and implements other models, or adjusting the models developed here. In addition, we would also recommend obtaining a larger sample, preferably a sample with a better balance of data if possible and apply an unsupervised learning method which can accommodate categorical variables which would allow us to better compare and contrast unsupervised against the supervised models. In addition, future studies would also benefit from incorporating more features. A major benefit of the application of improving our machine learning models for healthcare is the ability to reduce misdiagnosis by primary healthcare physicians, but without any data on the sensitivity and specificity of a diagnosis in office we are unable to comment if our model improves the detection rate and prevents error. Overall all our models produced would lead to possible delayed treatment and possible patient stress due to not obtaining 100% sensitivity and specificity. We were unable to find an appropriate acceptable level in the literature for error, though this would be disease specific and highlights that whilst machine learning could be applied to healthcare it should not be the only tool. This means these models should be used in conjunction with other tools such as consultation with a primary care physician, which could possibly lead to a reduction in misdiagnosis with primary care physicians, or incorporated into systems to flag possible oversight by a physician automatically.

Appendix & Code

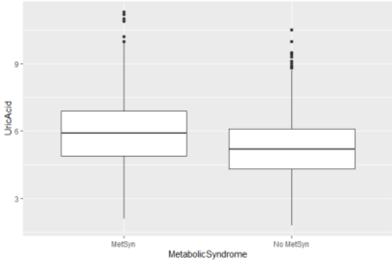


Figure 5: Performance of log transformation of UricAcid, allows us to better infer the relationship between the two classes.

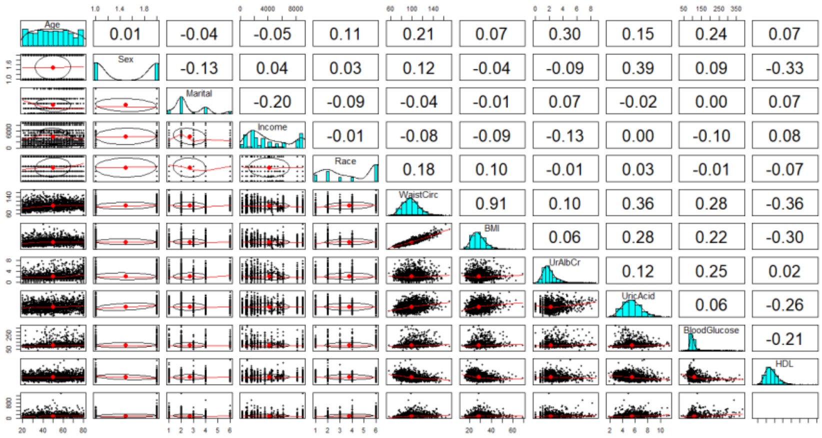
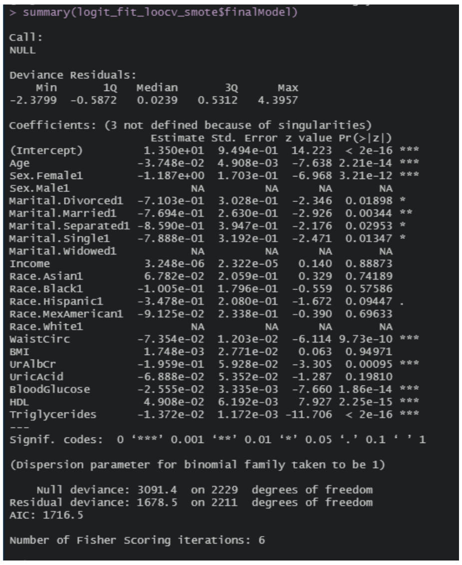


Figure 6: Correlation matrix on all features to test for Naïve Bayes assumption of independence. We define correlation of less then |0.39| as independent (Schober, Boer & Schwarte, 2018). This suggests that BMI & Waist circumfrance are not independent due to the high 0.91 correlation value.



```
> summary(logit_fit_loocv_smote$finalModel)

Call:
NULL

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3799  -0.5872   0.0239   0.5312   4.3957

Coefficients: (3 not defined because of singularities)
                      Estimate Std. Error z value Pr(>|z|)
(Intercept)          1.350e+01  9.494e-01  14.223  < 2e-16 ***
Age                 -3.748e-02  4.908e-03  -7.638 2.21e-14 ***
Sex.Female1         -1.187e+00  1.703e-01  -6.968 3.21e-12 ***
Sex.Male1                   NA         NA      NA       NA
Marital.Divorced1   -7.103e-01  3.028e-01  -2.346  0.01898 *
Marital.Married1    -7.694e-01  2.630e-01  -2.926  0.00344 **
Marital.Separated1  -8.590e-01  3.947e-01  -2.176  0.02953 *
Marital.Single1     -7.888e-01  3.192e-01  -2.471  0.01347 *
Marital.Widowed1            NA         NA      NA       NA
Income               3.248e-06  2.322e-05   0.140  0.88873
Race.Asian1          6.782e-02  2.059e-01   0.329  0.74189
Race.Black1         -1.005e-01  1.796e-01  -0.559  0.57586
Race.Hispanic1      -3.478e-01  2.080e-01  -1.672  0.09447 .
Race.MexAmerican1   -9.125e-02  2.338e-01  -0.390  0.69633
Race.White1                 NA         NA      NA       NA
WaistCirc           -7.354e-02  1.203e-02  -6.114 9.73e-10 ***
BMI                  1.748e-03  2.771e-02   0.063  0.94971
UrAlbCr             -1.959e-01  5.928e-02  -3.305  0.00095 ***
UricAcid            -6.888e-02  5.352e-02  -1.287  0.19810
BloodGlucose        -2.555e-02  3.335e-03  -7.660 1.86e-14 ***
HDL                  4.908e-02  6.192e-03   7.927 2.25e-15 ***
Triglycerides       -1.372e-02  1.172e-03 -11.706  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 3091.4  on 2229  degrees of freedom
Residual deviance: 1678.5  on 2211  degrees of freedom
AIC: 1716.5

Number of Fisher Scoring iterations: 6
```

Figure 7: Summary of coefficients produced for logistic model 4, this suggests that 3 variables have perfect multicollinearity as seen by NA coefficients. In addition, we observe 7 coefficients without a statically significant results suggesting we could further reduce dimensionality by omitting these features to improve our logit model.

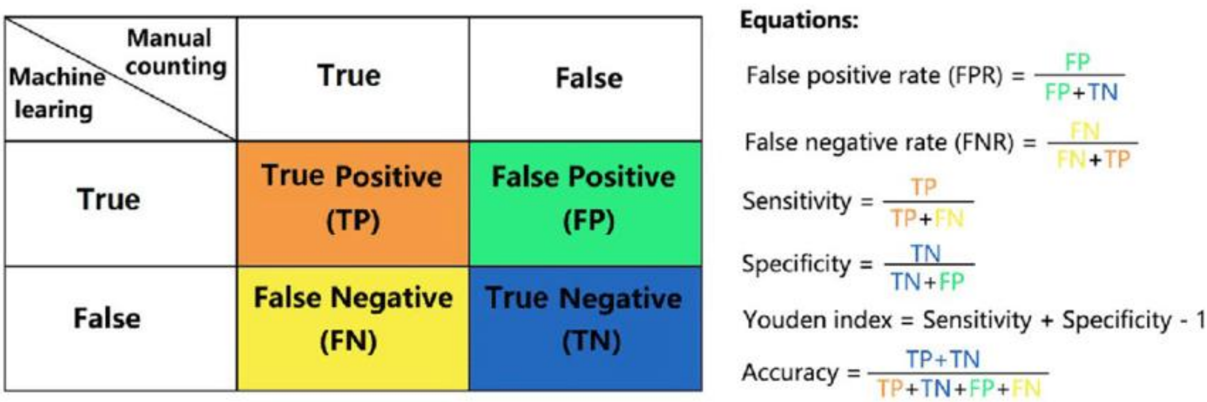


Figure 8: Method used to calculate accuracy, sensitivity and specificity from confusion matrix produced in unsupervised model (Niu et al., 2020).

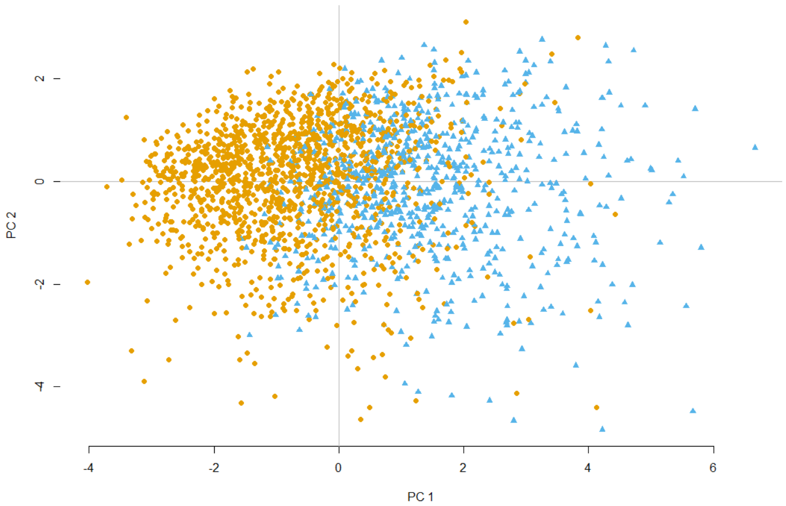


Figure 8: PCA plot of first two PC, suggesting that we have some separation visually between cluster types.

Install library===

library(VIM) # Used for imputation

library(MASS)

library(class)

#library(klaR)

library(dplyr)

library(pROC)

Set seed for reproducability - arbitrarily selected

```
set.seed(2343)


#======================================================================================
# Utilise various machine learning methods, both supervised and unsuperivsed
# to predict onces change of having metabolic syndrom; OR if metabolic syndrom
# clusters together.
#
#
# STEPS PROPOSED
# STEP 1:
# Data prep
# Check for nulls & decide on imputation
# Ensure each varible is converted to apporiate data type
# Create new data frames, one for supervised and one for unsupervised
# Implement first model - naive, LDA, QDA
# Implement second model - PCA
# Implement third model - Kmeans clustering



# https://www.frontiersin.org/articles/10.3389/fmed.2021.626580/full
# Similair studies have applied LDA/NB & Logstic regression to similair studies
# Based on other studies some important variables required to predict Mets are:
# WC, BMI, Obesity, DBp, SBp, Creatinine, Sex, UA, T_Billirbimum, Albumin, Escore
# CAPScore, TG, rGT, ALKp, GPT, GOT, HbA1C, GlucoseC, AFP, BUN, Age, TSH, HDL,
# MDRD, Cholesterol and LDL

# The above could suggest that categorical varibles such as income, maritial statues
# do not play a risk
```

```
#
https://www.cdc.gov/pcd/issues/2017/16_0287.htm#:~:text=When%20stratified%20by%20race%2Fethnicity,%E2%80%932012%20(Figure%202).

# The above study found that metabolic syndrom has increased amongts non-hispanic white women

# non-hispanic black women and people of low socioeconomic status


# The above suggests that we could also probably use Race and Income in our model(s)

# if assumptions allowed them to be included.


#===============================================================================




#===============================================================================

# DATA PREP - overal

#

#===============================================================================


metabolic_df <- read.csv("metabolic.csv")



# Ensure any characters are converted to NA

metabolic_df <- metabolic_df %>%

  mutate_all(na_if, "")


sum(is.na(metabolic_df)) # We have 436 varibles missing



sapply(metabolic_df, function(x) sum(is.na(x))) # Column wise summary of null values


#===============================================================================

# IMPUTATION
```

```
# https://towardsdatascience.com/7-ways-to-handle-missing-values-in-machine-learning-1a6326adf79e
# There are serveral methods we can employ to deal with various missing data
# Martial: We will imputate maritial if NULL -> unkown
# Income: Utilise VIM package kNN()
# Waist Circumstance: Utilise VIM package kNN()
#================================================================================

metabolic_df$Marital[is.na(metabolic_df$Marital)] <- "unknown"

# !!!
# There are some issues with imputation and machine learning, as such we
# will need to include this in disscusion section of paper
# !!!

metabolic_df_ni <- metabolic_df # Retain DF without k-means

#================================================================================
# IMPUTATION
# Income / Waist Circumfrance / BMI
#================================================================================

library(caret) # For imputation
imput_df <- metabolic_df %>%
  dplyr::select(-seqn  , -MetabolicSyndrome)

dummy_vars <- dummyVars(~ ., data = imput_df)
train_dummy <- predict(dummy_vars, imput_df)

pre_process <- preProcess(train_dummy, method = "bagImpute")
imputed_data <- as.data.frame(predict(pre_process, train_dummy))
```

```
metabolic_df$Income <- imputed_data$Income
metabolic_df$WaistCirc <- imputed_data$WaistCirc
metabolic_df$BMI <- imputed_data$BMI
sapply(metabolic_df, function(x) sum(is.na(x))) # Confrim imputation worked

#==============================================================================
# DATA TRANSFOMRATION
# Ensure data is of correct type
#==============================================================================

str(metabolic_df) # We observe many factor varibles are of type character

metabolic_df <- metabolic_df %>%
  mutate(Sex = as.factor(Sex),
         Marital = as.factor(Marital),
         Race = as.factor(Race),
         MetabolicSyndrome = as.factor(MetabolicSyndrome))
str(metabolic_df) # All varibles are now printing correctly



#==============================================================================
# DATA EXPLORATION
# Box plots
#==============================================================================
library(ggplot2)
# Look for outliers

(age_boxplot <- ggplot(metabolic_df, aes(x=MetabolicSyndrome, y=Age)) +
  geom_boxplot() ) # Different between groups
```

```
(income_boxplot <- ggplot(metabolic_df, aes(x=MetabolicSyndrome, y=Income)) +
   geom_boxplot() ) # Different between groups

(waist_boxplot <- ggplot(metabolic_df, aes(x=MetabolicSyndrome, y=WaistCirc)) +
   geom_boxplot() ) # Different between groups

(bmi_boxplot <- ggplot(metabolic_df, aes(x=MetabolicSyndrome, y=BMI)) +
   geom_boxplot() ) # Different between gorups

(urAlbCr_boxplot <- ggplot(metabolic_df, aes(x=MetabolicSyndrome, y=UrAlbCr)) +
   geom_boxplot() ) # Unable to gain any insight from this varible - transformation?

(uric_boxplot <- ggplot(metabolic_df, aes(x=MetabolicSyndrome, y=UricAcid)) +
   geom_boxplot() ) # Almost identical distribution - DROP - refer to literature, we do see some
outliers on higher end of MS & longer whisker on bottom end of no metsynd

(blood_glucos_boxplot <- ggplot(metabolic_df, aes(x=MetabolicSyndrome, y=BloodGlucose)) +
   geom_boxplot() ) # Lot of outliers

(hdl_boxplot <- ggplot(metabolic_df, aes(x=MetabolicSyndrome, y=HDL)) +
   geom_boxplot() ) # Higher in those without metabolic syndrome (good cholesterol )

(tri_boxplot <- ggplot(metabolic_df, aes(x=MetabolicSyndrome, y=Triglycerides)) +
   geom_boxplot() ) # Higher in those without metabolic syndrome

library(grid)
library(gridExtra)
grid.arrange(age_boxplot, income_boxplot, waist_boxplot, bmi_boxplot, urAlbCr_boxplot,
uric_boxplot, blood_glucos_boxplot, hdl_boxplot, tri_boxplot)

# Based on the boxplot the following transformation was conducted]
```

```
metabolic_df <- metabolic_df %>%
  mutate(UrAlbCr = log(as.integer(UrAlbCr)))
(uric_boxplot <- ggplot(metabolic_df, aes(x=MetabolicSyndrome, y=UricAcid)) +
   geom_boxplot() ) # Can now see almost identical distrubtion



(race_bar <- ggplot(metabolic_df, aes(x=Race)) +
   geom_bar() )
# We observe that a large portion of our data is: White, Black & Asian
# We are unsure the ethenicity of other, and since accounts for <10% of data
# we elected to remove it

(sex_bar <- ggplot(metabolic_df, aes(x=Sex)) +
   geom_bar() )



(relo_bar <- ggplot(metabolic_df, aes(x=Marital)) +
   geom_bar() )
# Previous studies have shown that relationship status can influence health status
# https://pubmed.ncbi.nlm.nih.gov/29976034/
# https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6166117/#:~:text=%5B6%5D%20have%20reported%20that%20women,physical%20and%20psychological%20support%2C%20especially



(alb_bar <- ggplot(metabolic_df, aes(x=Albuminuria)) +
   geom_bar() )
# Unsure what 0,1 & 2 refer to, might have to drop this variable if no data
# dict can be found - limitation of study and source data.
# Though, most obseervations are classed as 0 and could result in noise in our model
```

```
(ms_bar <- ggplot(metabolic_df, aes(x=MetabolicSyndrome)) +
   geom_bar() )

grid.arrange(race_bar, sex_bar, relo_bar, alb_bar, ms_bar)

#================================================================================
# REMOVE/CREATE VARS
# Since we do not know what other stands for we will remove it, also accounts
# for less then 10% of all observations
#================================================================================


n_row_initial <- nrow(metabolic_df)

metabolic_df <- metabolic_df %>%
  filter(Race != "Other") %>%
  filter(Marital !=  "unknown") %>%
  select(-Albuminuria) %>%
  unique() # Remove any duplicated rows - none detected anyhow

n_row_filter <- nrow(metabolic_df)
total_removed = n_row_initial - n_row_filter # 61 observations lost



#================================================================================
# DATA EXPLOR SUMMARY
# Based on exploration of data we observe most varibles have a clear difference
# between those with MS and those without. Some identified issues:
# Remove: Race (Other)
```

```
# Remove Column: Unsure what Albuinuria refers to - possible limitation - will review litrature to decide

# Transfomration: log UrAlbCr

# Unbalanced data: Metabolic syndrom unbalance - approx 800 MetSyn and approx 1540 No MetSyn

#================================================================================



#================================================================================

# Supervised data prep

#================================================================================

metabolic_df_supervised <- metabolic_df


#================================================================================

# Unsupervised data prep

# For our unsupervised learning method we will only select continuous varibles, at this

# stage we will retain seqn and metabolic syndrom status

#================================================================================


metabolic_df_unsupervised <- metabolic_df %>%

  select(#seqn,

        MetabolicSyndrome,

        Age,

        Income,

        WaistCirc,

        BMI,

        UrAlbCr,

        UricAcid,

        BloodGlucose,

        HDL,

        Triglycerides)
```

```
# Since we are dealing with units in different measurements we will apply
# a scaling algorithm prior to completing PCA analysis



#=================================================================================
# Testing significance
# We elected to use a non parametric test as we know from previous steps that our data
# has many outliers, normality is often violated etc.
# For numerical varibles we elected to do a Two sample Wilcoxon test for independent samples
#=================================================================================



metabolic_df_sig <- metabolic_df %>%
  select(-seqn)


# For p less then 0.05 we find evidence to support that varible is different between
# metabolic syndrome status
age_zw <- wilcox.test(Age ~ MetabolicSyndrome, data = metabolic_df_sig,
alternative="two.sided",mu=0) # P < 0.05
income_zw <- wilcox.test(Income ~ MetabolicSyndrome, data = metabolic_df_sig,
alternative="two.sided",mu=0) # P< 0.05
waist_zw <- wilcox.test(WaistCirc ~ MetabolicSyndrome, data = metabolic_df_sig,
alternative="two.sided",mu=0) # P < 0.05
BMI_zw <- wilcox.test(BMI ~ MetabolicSyndrome, data = metabolic_df_sig,
alternative="two.sided",mu=0) # P < 0.05
uric_a_zw <- wilcox.test(UricAcid ~ MetabolicSyndrome, data = metabolic_df_sig,
alternative="two.sided",mu=0) # P < 0.05
glucose_zw <- wilcox.test(BloodGlucose ~ MetabolicSyndrome, data = metabolic_df_sig,
alternative="two.sided",mu=0) # P < 0.05
HDL_zw <- wilcox.test(HDL ~ MetabolicSyndrome, data = metabolic_df_sig,
alternative="two.sided",mu=0) # P < 0.05
tri_zw <- wilcox.test(Triglycerides ~ MetabolicSyndrome, data = metabolic_df_sig,
alternative="two.sided",mu=0) # P < 0.05
```

```
# For categorical varibles we elected to do chi sqaure test

# Ho Two vars independent of one another in relation to MS status

# H1 Two vars are related to one another

chisq.test(y= metabolic_df_sig$MetabolicSyndrome, x=metabolic_df_sig$Sex) # P 0.351 - reject H1

chisq.test(y= metabolic_df_sig$MetabolicSyndrome, x=metabolic_df_sig$Marital) # P < 0.05 - reject
H0

chisq.test(y= metabolic_df_sig$MetabolicSyndrome, x=metabolic_df_sig$Race) # P < 0.05 - reject H0



#=================================================================================

# SUPERVISED LEARNING

# Naive Bayes - 4 different models

#=================================================================================


library(caret) # used to split data


#=================================================================================

# NAIVE BAYES: Model 1 & 2

#=================================================================================

set.seed(2343)

metabolic_df_supervised_nb <- metabolic_df_supervised %>%

  select(-seqn)


library(psych)

pairs.panels(metabolic_df_supervised_nb[,-13]) # Observe  cor feature <- assumption violation



split_nb <- createDataPartition(metabolic_df_supervised_nb$MetabolicSyndrome, p = 0.8, list = F)
```

```
train_nb <- metabolic_df_supervised_nb[split_nb, ]

test_nb <- metabolic_df_supervised_nb[-split_nb, ]

c(nrow(train_nb), nrow(test_nb)) # 1922 TRAIN, 479 TEST


# Check balance of data

prop.table(table(train_nb$MetabolicSyndrome)) # Data is somewhat imbalanced, 35% NMS, 65% MBS


# ASSUMPTION TESTING - Independence NB

library(psych)

pairs.panels(metabolic_df_supervised_nb)


# !!! We observe some varibles are correlated to one another - violation !!!


# DATA DISTRUBTUION - Gaussian Distribution

par(mfrow=c(2,2))

hist(metabolic_df$Age) # Not normal

hist(metabolic_df$Income) # Not normal

hist(metabolic_df$Triglycerides) # Skewed left

hist(metabolic_df$BloodGlucose) # Not normal

par(mfrow=c(1,1)) # Reset


# !!! We observe does not follow Gaussian Distrubtion, set kernel  = T


# Implement NAIVE BAYES

library(naivebayes)

nb_g <- naive_bayes(MetabolicSyndrome ~ ., data = train_nb, usekernel = F)

get_cond_dist(nb_g) # Observe distrubtion

nb_gf <- naive_bayes(MetabolicSyndrome ~ ., data = train_nb, usekernel = T)

get_cond_dist(nb_gf) # Observe distrubtion used for non paramtertic version

plot(nb_g, which = "BloodGlucose") # Test discriminatory power
```

```
# Produce confusion matrix for NAIVE BAYES

confusionMatrix(predict(nb_g), train_nb$MetabolicSyndrome)

confusionMatrix(predict(nb_gf), train_nb$MetabolicSyndrome)

# Based on test data we observe that the non-parametric model is slightly better


pred_g <- predict(nb_g, newdata = test_nb)

confusionMatrix(pred_g, test_nb$MetabolicSyndrome)


pred_gf <- predict(nb_gf, newdata = test_nb)

confusionMatrix(pred_gf, test_nb$MetabolicSyndrome)


# We observe that using kernal density plots in place of Gaussian improves our model

# this was expected based on the assumption testing


# NAIVE BAYES AURO

library(pROC)

roc_gaussian <- roc(as.numeric(test_nb$MetabolicSyndrome), as.numeric(pred_g)) # AUC 0.7264

roc_kernal <- roc(as.numeric(test_nb$MetabolicSyndrome), as.numeric(pred_gf)) # AUC 0.7526
(0.8336)



#==============================================================================

# NAIVE BAYES: Model 3

# Note we only use the non-gausian distribution in the report

#==============================================================================

# At this stage we elect to remove waist circumstance to see if it improves our model and

# to ensure all assumptions are satisfied



#metabolic_df_supervised_nb <- metabolic_df_supervised %>%
```

```
#  select(-seqn)

#metabolic_df_supervised_nb_2 <- metabolic_df_supervised_nb %>%
#  select(-BMI)

#split_nb_2 <- createDataPartition(metabolic_df_supervised_nb_2$MetabolicSyndrome, p = 0.8, list = F)
train_nb_2 <- train_nb %>%
  select(-BMI)
test_nb_2 <- test_nb %>%
  select(-BMI)

c(nrow(train_nb_2), nrow(test_nb_2))

# Produce models
nb_g_2 <- naive_bayes(MetabolicSyndrome ~ ., data = train_nb_2, usekernel = F)
nb_gf_2 <- naive_bayes(MetabolicSyndrome ~ ., data = train_nb_2, usekernel = T)

# Produce confusion matrix
pred_g_2 <- predict(nb_g_2, newdata = test_nb_2)
confusionMatrix(pred_g_2, test_nb_2$MetabolicSyndrome)

pred_gf_2 <- predict(nb_gf_2, newdata = test_nb_2)
confusionMatrix(pred_gf_2, test_nb_2$MetabolicSyndrome)

(roc_gaussian_2 <- roc(as.numeric(test_nb_2$MetabolicSyndrome), as.numeric(pred_g_2))) # AUC 0.7569
(roc_kernal_2 <- roc(as.numeric(test_nb_2$MetabolicSyndrome), as.numeric(pred_gf_2))) # AUC 0.8167
```

```
# Findings:

# AUC 0.85 remove BMI

# AUC 0.82 remove waist cir

# AUC 0.85 remove HDL

# AUC 0.82 remove Tri

# AUC 0.85 remove BMI and HDL



#================================================================================

# NAIVE BAYES DISC

# We observe that our assumptions have been violated, such as conditional independence

# amongst predictor variables, though Naive bayes can handle this generally

# We observe that our data is not Gausian Distrubtion - as such we can apply

# a non-paramatric feture in the naivebayes() function - this resulted in a

# improved model, both in accuracy and AUC

#================================================================================


#================================================================================

# NAIVE BAYES: Model 4

# Attempt to improve our model we use 10-fold cross validation and SMOTE

#================================================================================


ctrl <- trainControl(sampling = "smote")

nb_cv_smote <- train(MetabolicSyndrome ~ ., data = train_nb_2,

                     method = "nb",

                     trControl = ctrl)


print(nb_cv_smote)

pred_cv_smote <- predict(nb_cv_smote, newdata = test_nb_2)

confusionMatrix(pred_cv_smote, test_nb_2$MetabolicSyndrome)
```

```
roc_cv_smote <- roc(as.numeric(test_nb_2$MetabolicSyndrome), as.numeric(pred_cv_smote))

auc(roc_cv_smote) # 0.8221



#===============================================================================

# logit Supervised learning

# We retain BMI

# We test assumptions last - we will only test assumption on best

# performing model

#===============================================================================

set.seed(2343) # Ensure when you run model we run all of it with sed seed, otherwise it will not work if you do it in chunks.

library(caret)

# Convert categorical variables into dummy variables

metabolic_df_logit <- metabolic_df_supervised %>%

  select(-seqn, - MetabolicSyndrome)


diagnosis <- metabolic_df_supervised$MetabolicSyndrome


dummy_vars_logit <- dummyVars(~ ., data = metabolic_df_logit)

train_dummy_logit <- predict(dummy_vars_logit, metabolic_df_logit)

train_dummy_logit <- as.data.frame(train_dummy_logit)

train_dummy_logit <- cbind(train_dummy_logit, diagnosis)


# Bind diagnosis back to data frame with dummy varibles

#train_dummy_logit$diagnosis <- ifelse(train_dummy_logit$diagnosis == "No MetSyn", 0 , 1)

train_dummy_logit$diagnosis <- train_dummy_logit$diagnosis

str(train_dummy_logit)


# Convert variables to factor level
```

```
sum(train_dummy_logit$Marital.unknown) # 0, must be stored as memory as removed
sum(train_dummy_logit$Race.Other) # , must be stored as memory as removed before

train_dummy_logit <- train_dummy_logit %>%
  mutate(Sex.Female = as.factor(Sex.Female),
         Sex.Male = as.factor(Sex.Male),
         Marital.Divorced = as.factor(Marital.Divorced),
         Marital.Married = as.factor(Marital.Married),
         Marital.Separated = as.factor(Marital.Separated),
         Marital.Single = as.factor(Marital.Single),
         Marital.Widowed = as.factor(Marital.Widowed),
         Race.Asian = as.factor(Race.Asian),
         Race.Black = as.factor(Race.Black),
         Race.Hispanic = as.factor(Race.Hispanic),
         Race.MexAmerican = as.factor(Race.MexAmerican),
         Race.White = as.factor(Race.White),
         diagnosis = as.factor(diagnosis)) %>%
  select(-Marital.unknown, -Race.Other)
str(train_dummy_logit)

# Our data frame is now in the required format, we have set our binary dummy varibles to factors with 2 level
# and all other data types are correct
# !! We do assumption testing after our model is developed !!
# The only main assumption to test first is that our response (diagnosis) is binary, which it has been converted

split_logit <- createDataPartition(train_dummy_logit$diagnosis, p=0.8, list = F)
train_logit <- train_dummy_logit[split_logit, ]
test_logit <- train_dummy_logit[-split_logit, ]
c(nrow(train_logit), nrow(test_logit))
```

```
#================================================================================
# LOGIT MODEL: Basic
#================================================================================
logit_fit <- glm(diagnosis ~ ., data = train_logit,
              family = binomial(link = "logit"))

logs_odd_ratio <- predict(logit_fit, newdata = test_logit, type = "link")
proba <- predict(logit_fit, newdata = test_logit, type = "response")
pred_on_lodds <- ifelse(logs_odd_ratio > 0, "MetSyn", "No MetSyn")
pred_on_proba <- ifelse(proba > 0.5, "MetSyn", "No MetSyn")
all(pred_on_lodds == pred_on_proba)
confusionMatrix(as.factor(pred_on_proba), test_logit$diagnosis)

roc_basic <- roc(as.numeric(test_logit$diagnosis), as.numeric(as.factor(pred_on_proba)))
auc(roc_basic) # Terrible confusion matrix



#================================================================================
# LOGIT MODEL: Leave one out
#================================================================================

# Cross validation: Leave one out cross validation
ctrl <- trainControl(method = "LOOCV")
logit_fit_loocv <- train(diagnosis ~ ., data = train_logit,
                   method = "glm",
                   family = "binomial",
                   trControl = ctrl)

pred_loocv <- predict(logit_fit_loocv, newdata = test_logit)
confusionMatrix(pred_loocv, test_logit$diagnosis)
```

```
roc_loocv <- roc(as.numeric(test_logit$diagnosis), as.numeric(pred_loocv))
auc(roc_loocv) # 0.84



#===============================================================================
# LOGIT MODEL: Cross Validation 10-fold
#===============================================================================
ctrl <- trainControl(method = "repeatedcv", number = 10,
                     savePredictions = TRUE, repeats = 5)
logit_fit_cv <- train(diagnosis ~ ., data = train_logit,
                      method = "glm",
                      family = "binomial",
                      trControl = ctrl)
print(logit_fit_cv)
pred_cv <- predict(logit_fit_cv, newdata = test_logit)
confusionMatrix(pred_cv, test_logit$diagnosis)

roc_cv <- roc(as.numeric(test_logit$diagnosis), as.numeric(pred_cv))
auc(roc_cv) # 0.84

#===============================================================================
# Based on the intial logit models we observe that we might have an issue
# with unblanced nature impacting our model - we will now compare and contrast
# by including smote into our model
# Could not find ROSE used in the context of health science.
#
# Based on confusion matrix we observe LOOCV had best outcome, thus we will
# see if we can improve it by accounting for an unblaanced data set
#===============================================================================
```

```
#================================================================================
# LOGIT MODEL: Leave one out - SMOTE applied
#================================================================================


ctrl <- trainControl(method = "LOOCV", sampling = "smote")
logit_fit_loocv_smote <- train(diagnosis ~ ., data = train_logit,
                    method = "glm",
                    family = "binomial",
                    trControl = ctrl)

pred_loocv_smote <- predict(logit_fit_loocv_smote, newdata = test_logit)
confusionMatrix(pred_loocv_smote, test_logit$diagnosis)

roc_loocv_smote <- roc(as.numeric(test_logit$diagnosis), as.numeric(pred_loocv_smote))
auc(roc_loocv_smote)

summary(logit_fit_loocv_smote$finalModel)

#================================================================================
# FINDINGS:
# We find when we apply the SMOTE method for LOOCV that we have an increase
# in false positives - this results in a reduction in our AUC slightly
# though, we have an increase by 0.10 in sensitivity and a 0.9 reduction in
# specificity - we also have a reduction in accuracy in our model.
# This is because our model has a reduction in original biased towards no met
# syndrome.
# For the purpose of health applications we want to maxmimise sensitivity
# as failure to do so can lead to delayed treatment
#================================================================================
```

```
#===============================================================================
# Logit assumption testing
# 3 tests:
# Multicollinearity
# linearity
# outliers
#273
#===============================================================================

#===============================================================================
# Multicollinearity test
#===============================================================================

library(car)
vif(logit_fit_loocv_smote$finalModel) # Produces error



# VIF error suggests we have perfect multi-collinearity in our model due to error type
# https://stats.stackexchange.com/questions/495702/r-dealing-with-new-aliased-coefficient-na-coefficient-in-categorical-varia
summary(logit_fit_loocv_smote)

# We observe that our dummy variables are cuasing perfect multicolinearity
# Sex.Male
# Marital.Widow
# Race.White

# To overcome this issue we create a new subsset of test/train by removing those
# observations and re-running our model with the reduced test/train data frame
# all else held constent - thus producing a 5th model essentially - this is because
```

```
# when the others are equal to 0 it is implied that they are either male, widows or white



train_logit_2 <- train_logit %>%
  select(-Sex.Male, -Marital.Widowed, -Race.White)


test_logit_2 <- test_logit %>%
  select(-Sex.Male, -Marital.Widowed, -Race.White)


ctrl <- trainControl(method = "LOOCV", sampling = "smote")
logit_fit_loocv_smote_2 <- train(diagnosis ~ ., data = train_logit_2,
                                 method = "glm",
                                 family = "binomial",
                                 trControl = ctrl)


pred_loocv_smote_2 <- predict(logit_fit_loocv_smote_2, newdata = test_logit_2)
confusionMatrix(pred_loocv_smote_2, test_logit_2$diagnosis)


roc_loocv_smote_2 <- roc(as.numeric(test_logit_2$diagnosis), as.numeric(pred_loocv_smote_2))
auc(roc_loocv_smote_2)


vif(logit_fit_loocv_smote_2$finalModel) # Multicolineairty detected amongsst BMI and Waist Circumference, only slightly VIF > 5 but VIF<6
summary(logit_fit_loocv_smote_2)


#=============================================================================
# Outliers
#=============================================================================
plot(logit_fit_loocv_smote_2$finalModel, which = 4, id.n=3) # Cooks distance


library(broom) # Required for augmnet
```

```
model_outliers <- augment(logit_fit_loocv_smote_2$finalModel) %>%
  mutate(index = 1:n())



residual_plot <- ggplot(model_outliers, aes(index, .std.resid)) +
  geom_point(aes(color = .outcome), alpha = .5) +
  theme_bw() # Visually looks like all less then 3

model_outliers <- model_outliers %>%
  filter(abs(.std.resid) > 3)

nrow(model_outliers) # We have 7 observations which we could consider an outlier
max(abs(model_outliers$.std.resid)) # Max .abs std residual is 4.31

#==============================================================================
# Linearity
# Issue due to:
# Because of SMOTE method we have extra values included - as we have synthetically
# inserted these values. My skills are unable to remove these from the
# logit_fit_loocv_smote_2$finalModel, instead we will use logit_fit_loocv
# just to look at linearity
#==============================================================================
library(tidyverse)

probabilities <- predict(logit_fit_loocv$finalModel, type = "response")
predicted.classes <- ifelse(probabilities > 0.5, "MetSyn", "No MetSyn")

# Select only numeric predictors
dt <- train_logit_2 %>%
  dplyr::select_if(is.numeric)
predictors <- colnames(dt)
```

```
dt <- dt %>%
  mutate(logit = log(probabilities / (1 - probabilities))) %>%
  gather(key = "predictors", value = "predictor_value", -logit)


linear_plot <- ggplot(dt, aes(logit, predictor_value)) +
  geom_point(size = .5, alpha = .5) +
  geom_smooth(method = "loess") +
  theme_bw() +
  facet_wrap(~predictors, scales = "free_y")

# We observe that linearity assumption is most likely violated in our model
library(gridExtra)
grid.arrange(linear_plot, residual_plot, nrow=1)




#==============================================================================
# Unsupervised learning:
# Goal: 2 models, 1 with PCA applied and 1 without PCA applied
#==============================================================================
set.seed(2343)

ncol(metabolic_df_unsupervised) # 9 features, reduction in previous model due to continous assumption

metabolic_df_unsupervised_diagnosis <- as.data.frame(metabolic_df_unsupervised$MetabolicSyndrome) # Retain to test model at later stage
```

```
metabolic_df_unsupervised_model_1 <- metabolic_df_unsupervised %>%
  select(-MetabolicSyndrome) # Unblabel our model


metabolic_df_unsupervised_model_1_scaled <- scale(metabolic_df_unsupervised_model_1) # Scale data, but retain orginal


k2 <- kmeans(metabolic_df_unsupervised_model_1_scaled, centers = 2, nstart = 25)
k2$size
cluster <- as.data.frame(k2$cluster)


library(factoextra)
fviz_cluster(k2, data = metabolic_df_unsupervised_model_1_scaled) # Graphs first 2 PC, observe clear separation
fviz_nbclust(metabolic_df_unsupervised_model_1_scaled, kmeans, method = "silhouette") # Observe k=2 as recomended - based on domain knoweldge


# Interpret our results
library(data.table)
dt_ext <- as.data.table(cbind(metabolic_df_unsupervised_model_1, cluster = k2$cluster))
dt_ext[, lapply(.SD, mean), by = cluster]


bind_metabolic_df_unsupervised_model_1_scaled <- cbind(metabolic_df_unsupervised_diagnosis, cluster) # Allows us to see cluster pred vs actual diagnosis
(cont_table <- table(bind_metabolic_df_unsupervised_model_1_scaled))
accuracy_model_1 <- sum(diag(cont_table)/sum(cont_table))  # Accuracy 0.7946679
(summary_stats_model <- dt_ext[, lapply(.SD, mean), by = cluster])


#==============================================================================
# Disscusion:
# Based on contingency table above - we assume that cluster 1 is MetSyn, whilst cluster 2 is
# No MetSyn, this produces an accuracy of 79%
# We are also able to obtain some summary stats, we observe people in cluster 1 have the\
```

following charactersitics:

Higher age, lower income, higher waist Cir, higher BMI, higher UrAlAbrc, higher UricAcid

higher blood glucose, lower HDL and higher Triglycerdies - this is consisent with the litrature

#==

Question: Can we improve our model via the application of PCA first, then apply k-means clustering

NOTE: With this method we will loose descriptive infromation such as summary stats, though

we can still produce accuracy etc

pca_model <- prcomp(metabolic_df_unsupervised_model_1_scaled, scale = F) # Already scaled

summary(pca_model) # approx 80% var explained by first 5 PC

PVE_model <- round((PVE_model <- (pca_model$sdev^2)/sum(pca_model$sdev^2)),2) # Round our variance

Shows prop of variance explained by each PCA

par(mfrow = c(1,2))

plot(PVE_model, xlab = "Principal Component", ylab = "Prop of variance Explained", type = "b", ylim = c(0,1))

Shows prop of variance explained cumualaivte of each PCA as we go through list

plot(cumsum(PVE_model), xlab = "Principal component", ylab = "cummulative prop of variance Explained", type = "b")

par(mfrow = c(1,1)) # Reset

psych::pairs.panels(pca_model$x)

biplot(pca_model, scale = 0, cex = .5, col = c("grey", "deeppink3")) # Make observations grey for ease read

?biplot

Elbow method selecting number PC to use

fviz_eig(pca_model) # select first 2/3 PC values based on elbow

library(pca3d)

```
diagnosis_group <- as.factor(metabolic_df_unsupervised$MetabolicSyndrome) # For 3d plot below

pca3d(pca_model, group = diagnosis_group, legend = "topleft") # 3d graph - can see some clear seperation, model may struggle with middle values

pca2d(pca_model, group = diagnosis_group, title = "d", legend = "topleft")


# Look at roation

loading_pc1 <- pca_model$rotation[,1]

loading_pc2 <- pca_model$rotation[,2]

loading_pc3 <- pca_model$rotation[,3]

loading_pc1_abs <- abs(loading_pc1)

loading_pc2_abs <- abs(loading_pc2)

loading_pc3_abs <- abs(loading_pc3)

print(head(sort(loading_pc1_abs, decreasing = T)) )

print(head(sort(loading_pc2_abs, decreasing = T)) )

print(head(sort(loading_pc3_abs, decreasing = T)) )



pca_model <- pca_model$x[,1:3] # select PC1, PC2, PC3


k2_pc <- kmeans(pca_model, centers = 2, nstart = 25)

pc_cluster <- as.data.frame(k2_pc$cluster)

dt_ext_2 <- as.data.table(cbind(metabolic_df_unsupervised_model_1, cluster = k2_pc$cluster))

dt_ext_2[, lapply(.SD, mean), by = cluster]

bind_pc <- cbind( metabolic_df_unsupervised_diagnosis, pc_cluster)

(cont_table_pc <- table(bind_pc))

(accuracy_model_2 <- sum(diag(cont_table_pc)/sum(cont_table_pc))) # Accuracy 0.795136 - minor improvement with PCA applied
```

```
#================================================================================
# PCA UNSUPERVISED
#
# Will utilise PCA to reduce dimensionality of our data and apply it to our
# clustering technique
#================================================================================

# Remove seqn and MetabolicSyndrom
metabolic_df_unsupervised_pca <- metabolic_df_unsupervised %>%
  select( -MetabolicSyndrome)

# Scale as units are diff measurements/values
metabolic_df_unsupervised_pca <- as.data.frame(scale(metabolic_df_unsupervised_pca))
#metabolic_df_unsupervised_pca <- as.data.frame((metabolic_df_unsupervised_pca))



summary(metabolic_df_unsupervised_pca) # Confrim scaled
diagnosis <- as.numeric(metabolic_df_unsupervised$MetabolicSyndrome == "MetSyn") # MetSyn =
1, NO METSYN = 0

# Commence PCA and obtain some info
pca <- prcomp(metabolic_df_unsupervised_pca, scale = F) # Already scaled
summary(pca) # Summary PC values
PVE <- round((pca$sdev^2)/sum(pca$sdev^2), 2)

# Scree Plot
par(mfrow = c(1,1))
plot(PVE, xlab = "Principal Component", ylab = "Prop of variance Explained", type = "b", ylim = c(0,1))
```

```
# Shows prop of variance explained cumualaivte of each PCA as we go through list
plot(cumsum(PVE), xlab = "Principal component", ylab = "cummulative prop of variance Explained",
type = "b")

# Elbow method select PC values for model
library(factoextra)
fviz_eig(pca) # Select first 3 PC values for our model

# Show visually our model - we can see some speration between our model
library(pca3d)
pca2d(pca, group = diagnosis)

pca_3d_graph <- pca3d(pca, group=diagnosis, show.ellipses=TRUE,
                ellipse.ci=0.95, show.plane=FALSE, legend = "topleft")


# What varibles contribute most to our PCA model?
loading_pc1 <- abs(pca$rotation[,1])
loading_pc2 <- abs(pca$rotation[,2])
print(head(sort(loading_pc1, decreasing = T)))
print(head(sort(loading_pc2, decreasing = T)))

#================================================================================
# What the loading tells us is the most impact a var has on the PC
# PCA1: WaistCirc > BMI > HD: > Tri > Blood Gluc > UricAcid
# PCA2: UrAlbCr > Age > Blood Glucose > HDL > BMI > Income
#================================================================================

#================================================================================
# K-means clustering
```

```
#================================================================================

metabolic_df_unsupervised_clustering <- pca$x[,1:2] # Select first 2 PC

metabolic_df_unsupervised_clustering <-
as.data.frame(cbind(metabolic_df_unsupervised_clustering, diagnosis)) # MetSyn = 1, NO METSYN =
0

k2 <- kmeans(metabolic_df_unsupervised_clustering[1:2], centers = 2, nstart = 25) # We know k=2
domain knowledge


metabolic_df_unsupervised_clustering$cluster <- factor(k2$cluster)


# Produce table

table <- metabolic_df_unsupervised_clustering %>%

  mutate(diagnosis = case_when(diagnosis == 1 ~ "MetSyn",

                               diagnosis == 0 ~ "No MetSyn"))


cont_table_kmeans <- (table(table$diagnosis, table$cluster))

(accuracy_kmeans <- sum(diag(cont_table_kmeans)/sum(cont_table_kmeans)))


#================================================================================

# Obtained accuracy of 77% via unsupervised k-means clustering with PCA

# This is on par with supervised method

#================================================================================
```

Reference

Artificial Intelligence in Healthcare Market worth $67.4 billion by 2027 - Exclusive Report by MarketsandMarkets™. (2021). Retrieved 18 April 2022, from https://www.prnewswire.com/news-releases/artificial-intelligence-in-healthcare-market-worth-67-4-billion-by-2027--exclusive-report-by-marketsandmarkets-301411884.html

Ding, C., & He, X. (2004). K means clustering via principal component analysis. Twenty-First International Conference On Machine Learning - ICML '04. doi: 10.1145/1015330.1015408

Hoyt, R. (2019). Metabolic Syndrome Prediction - dataset by informatics-edu. Retrieved 25 April 2022, from https://data.world/informatics-edu/metabolic-syndrome-prediction

Kim, J., Mun, S., Lee, S., Jeong, K. and Baek, Y., 2022. Prediction of metabolic and pre-metabolic syndromes using machine learning models with anthropometric, lifestyle, and biochemical factors from a middle-aged population in Korea. BMC Public Health, 22(1).

Karabiber, F., 2022. Dummy Variable Trap – LearnDataSci. [online] Learndatasci.com. Available at: <https://www.learndatasci.com/glossary/dummy-variable-trap/> [Accessed 25 April 2022].

Moore, J., Chaudhary, N. and Akinyemiju, T., 2017. Metabolic Syndrome Prevalence by Race/Ethnicity and Sex in the United States, National Health and Nutrition Examination Survey, 1988–2012.

Niu, J., An, G., Gu, Z., Li, P., Liu, Q., & Bai, R. et al. (2020). Analysis of sensitivity and specificity: precise recognition of neutrophils during regeneration of contused skeletal muscle in rats. Forensic Sciences Research, 1-10. doi: 10.1080/20961790.2020.1713432

Publichealth.columbia.edu. 2022. K-Means Cluster Analysis | Columbia Public Health. [online] Available at: <https://www.publichealth.columbia.edu/research/population-health-methods/k-means-cluster-analysis> [Accessed 23 April 2022].

R - dealing with new aliased coefficient (“NA” coefficient) in categorical variables for VIF. (2020). Retrieved 22 April 2022, from https://stats.stackexchange.com/questions/495702/r-dealing-with-new-aliased-coefficient-na-coefficient-in-categorical-varia

Ramezankhani, A., Pournik, O., Shahrabi, J., Azizi, F., Hadaegh, F. and Khalili, D., 2014. The Impact of Oversampling with SMOTE on the Performance of 3 Classifiers in Prediction of Type 2 Diabetes. Medical Decision Making, 36(1), pp.137-144.

Singh, H., Schiff, G., Graber, M., Onakpoya, I., & Thompson, M. (2016). The global burden of diagnostic errors in primary care. BMJ Quality &Amp; Safety, 26(6), 484-494. doi: 10.1136/bmjqs-2016-005401

Schober, P., Boer, C., & Schwarte, L. (2018). Correlation Coefficients. Anesthesia &Amp; Analgesia, 126(5), 1763-1768. doi: 10.1213/ane.0000000000002864

Yu, C., Chang, S., Lin, C., Lin, Y., Wu, J. and Chen, R., 2021. Identify the Characteristics of Metabolic Syndrome and Non-obese Phenotype: Data Visualization and a Machine Learning Approach. Frontiers in Medicine, 8.