Select **one dataset in health and biomedicine** of your choice and appropriately **apply two supervise machine learning models** from the course

---

1) Introduction

Under the investigation of this machine learning study, the Cleveland heart disease dataset (Detrano et al. 1989) provides an important opportunity to utilise data driven insights into the diagnosis of Coronary artery disease (CAD). This current study intends to use the the Cleveland heart disease dataset to compare and evaluate the use of a k-nearest neighbours (kNN) and support vector machine (SVM) supervised machine learning classifier as a mechanism of predicting whether a patient presents with a diagnosis of CAD or is otherwise healthy.

In light of related literature, CAD is a leading cause of cardiovascular(CVD) deaths across high, middle low income countries (Sanchis-Gomar et al. 2016), characterised by greater than 50% narrowing of one of the three major coronary arteries. Doing more to diagnosis CAD appropriately and early is of regular active research. The current gold standard for the diagnosis of CAD is achieved using a coronary angiography (Lim & White 2013). Although accurate, the coronary angiography is an invasive procedure and relatively expensive (Van Mieghem 2017). Datasets like that of the Cleveland heart disease dataset (under analysis as part of this study) coupled with increasingly successful application of machine learning techniques in healthcare (Wiens & Shenoy 2018), offers an opportunity to design a potential algorithmic tool for supporting the diagnosis of CAD.

Compared to the other heart disease databases, the Cleveland heart disease dataset is the only one out of the four other databases in the heart disease repository that has been used by machine learning researchers and therefore allows further comparison of results during our discussion below. The Cleveland dataset contains 303 instances and 14 features for which are described further in **Table 1.** Our attribute of prediction is a feature named: *num* – diagnosis of heart disease based on the angiographic narrowing status of coronary arteries. The purpose of this machine learning study therefore is to evaluate application of two separate machine learning models in order to appropriately predict the presence of CAD – based of a predication of a >50% diameter narrowing of coronary arteries. The remaining 13 features as part of the dataset are related to measures that might help to describe the condition of an individual's heart.

Table 1 – Cleveland heart disease dataset features, description and class labels

Continuous features		
Feature name	**Feature description**	
age	Age (in years)	
trestbps	Resting blood pressure (in mm Hg on admission to the hospital)	
chol	Serum cholesterol (in mg/dl)	
thalach	Maximum heart rate achieved	
oldpeak	ST depression induced by exercise relative to rest	
Categorical features		
Feature name	**Feature description**	**Class labels**
sex	Sex of individual	- 0: Female - 1: Male
cp	Chest pain type	- 1: typical angina - 2: atypical angina - 3: non-anginal pain - 4: asymptomatic
restecg	Resting electrocardiographic results	- 0: normal - 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV) - 2: showing probable or definite left ventricular hypertrophy by Estes' criteria
exang	Exercise induced angina	- 0: No - 1: Yes
slope	The slope of the peak exercise ST segment	- 1: Upsloping - 2: Flat - 3: Downsloping
ca	Number of major vessels (0-3) coloured by fluoroscopy	- 1: 1 vessel coloured - 2: 2 vessels coloured - 3: 3 vessels coloured
thal	Thallium scintigram – radio isotype test used in the assessment of coronary artery disease.	- 3: Normal - 6: Fixed defect - 7: Reversible defect
num	Diagnosis of heart disease (angiographic disease status)	- 0: < 50% diameter narrowing - 1: > 50% diameter narrowing

Descriptive statistics

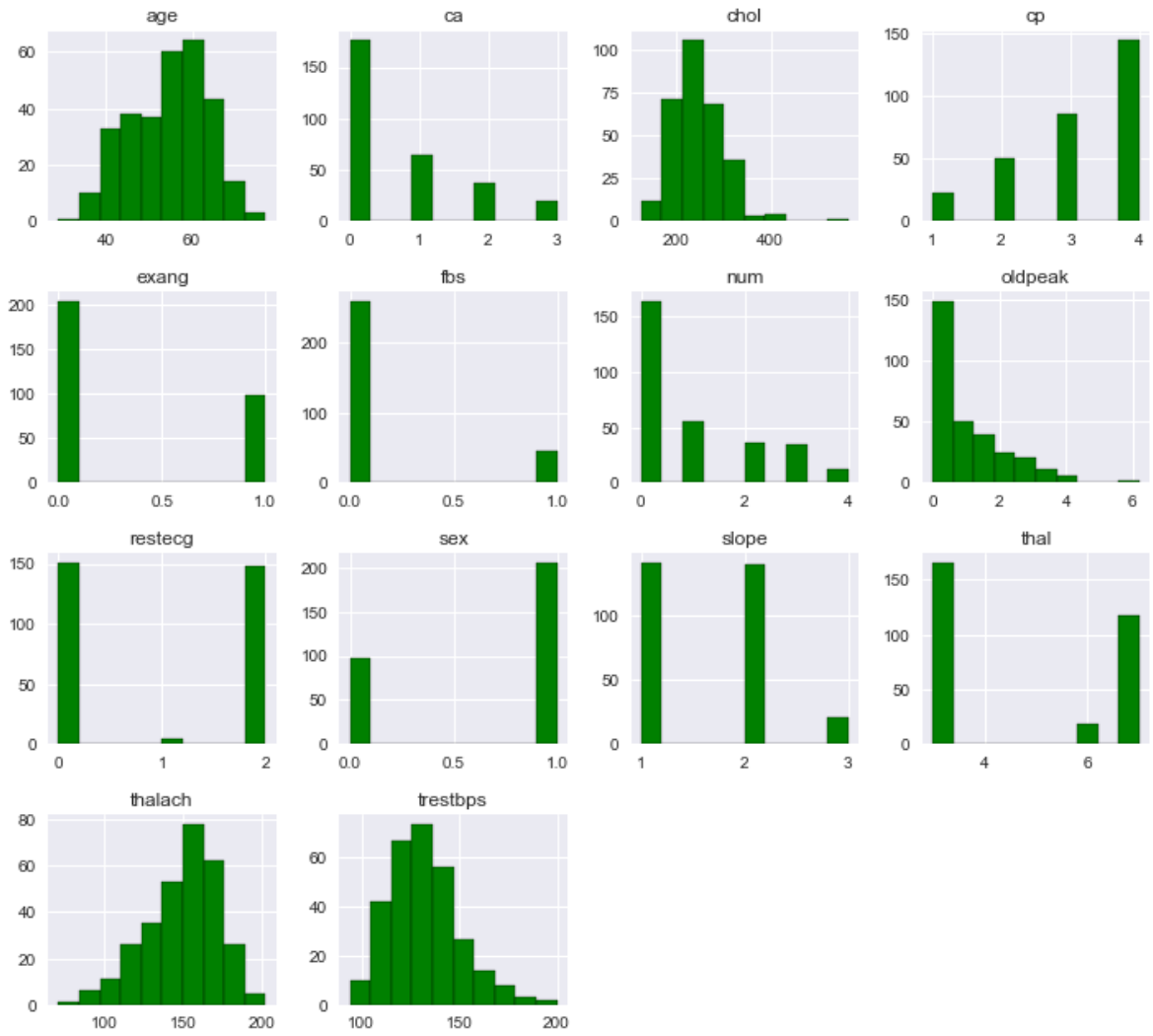
Initial analysis of our feature set suggests that were six potential missing values from our dataset. Likely to be associated with two missing values from *ca* and four from *thal*. To allow for complete descriptive statistical analysis of our dataset, all values of feature set were made numerical and missing data previously labelled as '?' recoded as 'NaN'. In terms of demographics, our dataset population has a mean age of 54 years old, a minimum age of 29 and maximum age of 77. Our dataset population also appears to present with over 100 more men than women. Although skewed these difference in demography may be appropriate as a means of observing a reasonably fair balance of healthy and CAD instances. For our prediction variable once re-categorised into values {0} and {1}. There were 164 instances of < 50% diameter narrowing and 139 instances of > 50% diameter narrowing. For further analysis of descriptive statistics for the Cleveland dataset, refer to **Table 2** below.

Table 2 – Descriptive statistics of the Cleveland dataset

Continuous features								
Feature name	**Count**	**Mean**	**Std**	**Min**	**25%**	**50%**	**75%**	**Max**
age	303	54.44	9.04	29	48	56	61	77
trestbps	303	131.69	17.60	94	120	130	140	200
chol	303	246.69	51.78	126	211	241	275	564
thalach	303	149.61	22.88	71	133.5	153	166	202
oldpeak	303	1.04	1.16	0	0	0.8	1.6	6.2
Categorical features								
sex	303	0.68	0.47	0	0	1	1	1
cp	303	3.16	0.96	1	3	3	4	4
restecg	303	0.99	0.99	0	0	1	2	2
exang	303	0.33	0.47	0	0	0	1	1
slope	303	1.60	0.62	1	1	2	2	3
ca	299	0.67	0.94	0	0	0	1	3
thal	301	4.73	1.94	3	3	3	7	7
num	303	0.94	1.23	0	0	0	2	4

Using histograms pictured in **Figure 1** to visualise our dataset, it becomes clear that 11 out of the 13 features are slightly or significantly skewed in distribution. The following variables can be strongly considered skewed to the left: *ca, chol, exang, fps, slope, trestbps and oldpeak.* Whilst the following few variables might strongly be considered skewed to the right: *age, cp, sex, and thalach*.

Figure 1 – Histograms of the Cleveland dataset



Visualising the feature space

On review of our correlation matrix (**Figure 2**) we are able to investigate the dependence between multiple variables within our dataset. The positive highest correlating variables to our predicted attribute (*num*) include *thal, ca, oldpeak* and *exang*. Using the 3D feature space (**Figure 3**) visualisation of how highest correlating features we are able to see how those with (coloured in red) and without a heart disease condition (coloured in blue) are distributed with respective to our highly positive correlating features *thal, ca and oldpeak*.

Figure 2 – Correlation matrix of the Cleveland dataset

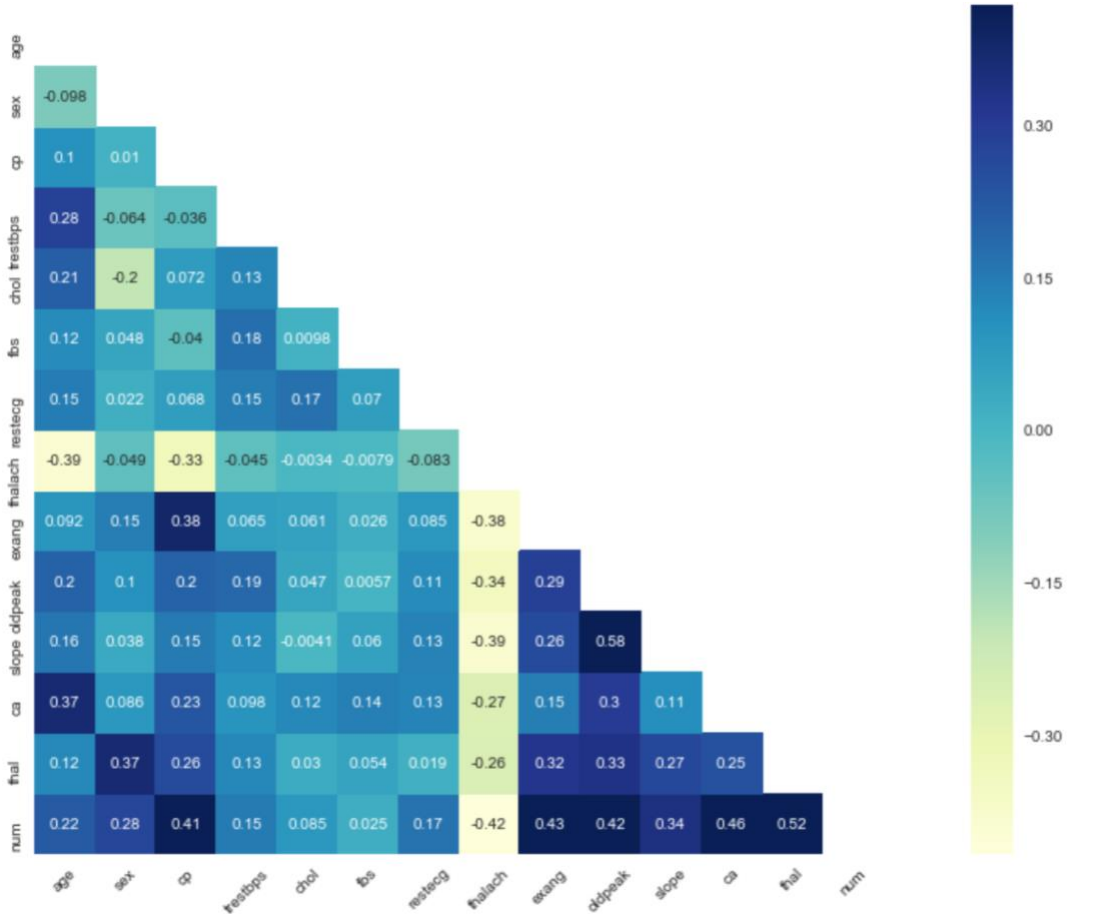
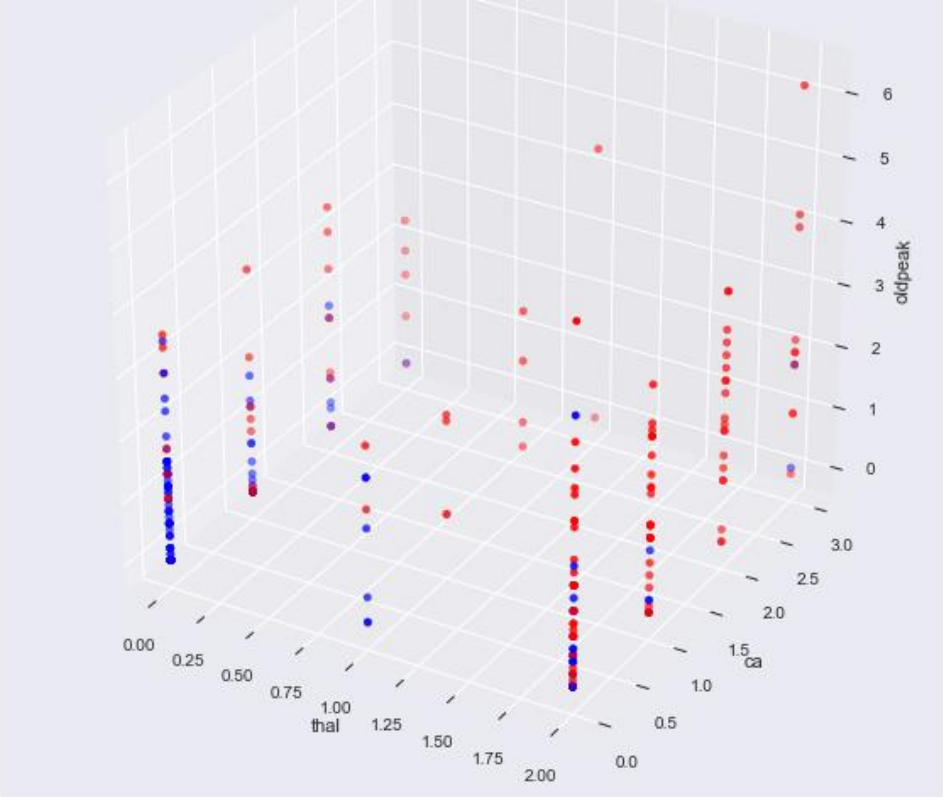


Figure 3 – 3D features space of the thal, ca and oldpeak features



2) Methodology

Our methodology for creating our classifier for CAD involves four stages: Data pre-processing, Model selection, Model training, and finally model evaluation.

Data pre-processing

Data cleaning

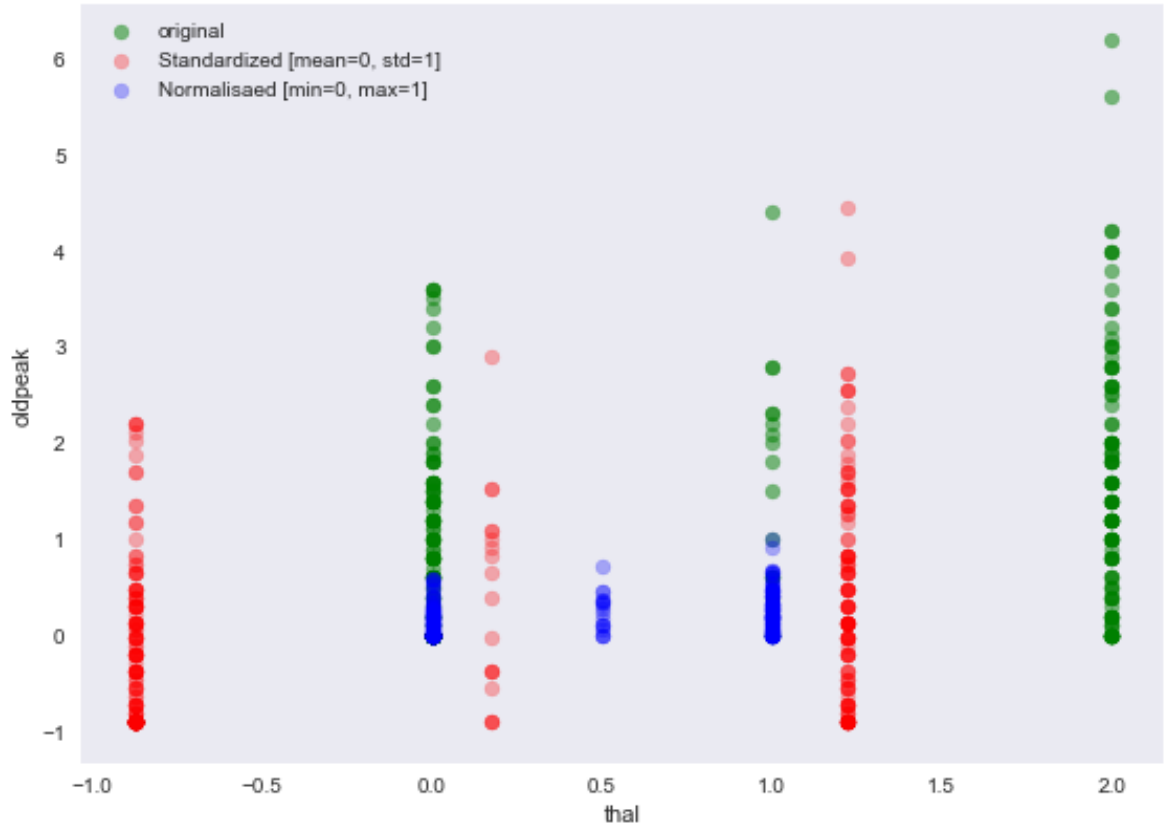
Once missing data had been numerically converted as described above. The missing values for features: *ca* and *thal* were imputed using the median value of the column. Both *ca* and *thal* are categorical variables and so the median class presents as the more appropriate mechanism of imputation than to identify a mean value for which is likely not to fall directly into a class value.

As part of further data cleaning, each categorical variable that hadn't already been labelled as such, would be relabelled so that each class began with {0} and would increase consecutively. This is encourage consistency between class labels.

Feature pre-processing

All our features do not exist along the same scale or magnitude. Therefore considering strategies of presenting our features in a way that is comparable is important. Process of normalisation and standardisation are two techniques considered as part our strategy for feature pre-processing. **Figure 4**, presents the effect of normalisation and standardisation on two of our features – *oldpeak* and *thal*. 11 of our features as described above do not follow a standard normal distribution and are skewed slightly or significantly to one side. Taking this into account normalisation seems a more valuable means of processing our features for supervised learning. But the impact on accuracy of our model for doing so will be analysed below as part of our mechanism for feature selection. Finally as part of pre-processing, our data is split into 70%` for training and 30% for testing.

Figure 4 – 2D feature space of the original, normalised and standardised data for *oldpeak* and *thal*



Feature selection

As part of creating an appropriate classifier for CAD, it is important to understand the value of each feature used within our model. It is important to include features that add significantly to the accuracy or variance of our model and remove those that do not. All with the intention of creating a model that reaches a useful balance between the extremes of over and under fitting. Sequential backwards selection with an untuned kNN classifier is used to help identify the most relevant features for further model training. This test of accuracy was initiated with the original feature set (**Figure 5**) but also with our feature set once standardised (**Figure 6**) and then also separately normalised (**Figure 7**). The process of doing so is to further add evidence to the most useful prep-processing technique that should be included as part of our pipeline.

Figure 5 – Results of the sequential backwards selection using original feature set

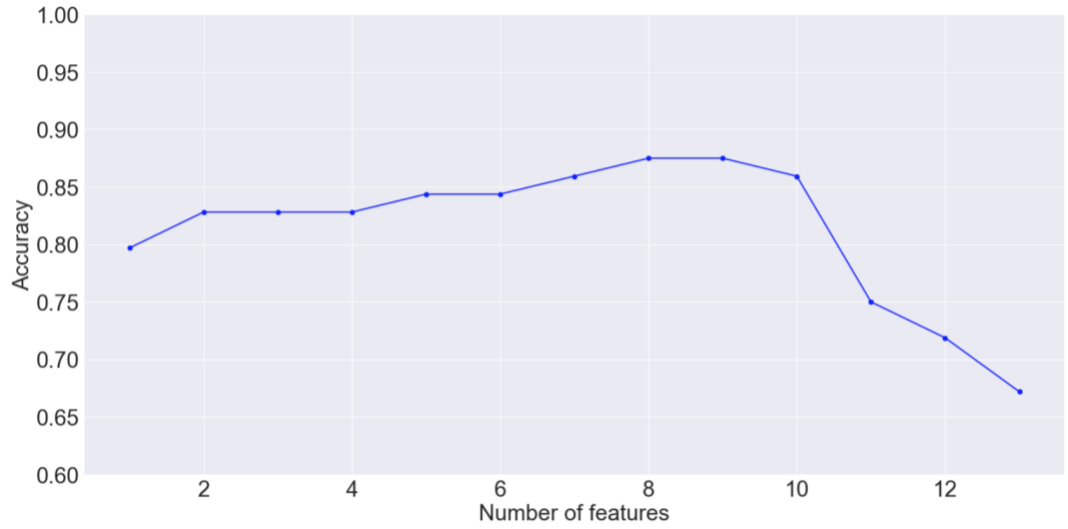


Figure 6 – Results of the sequential backwards selection using standardised feature set

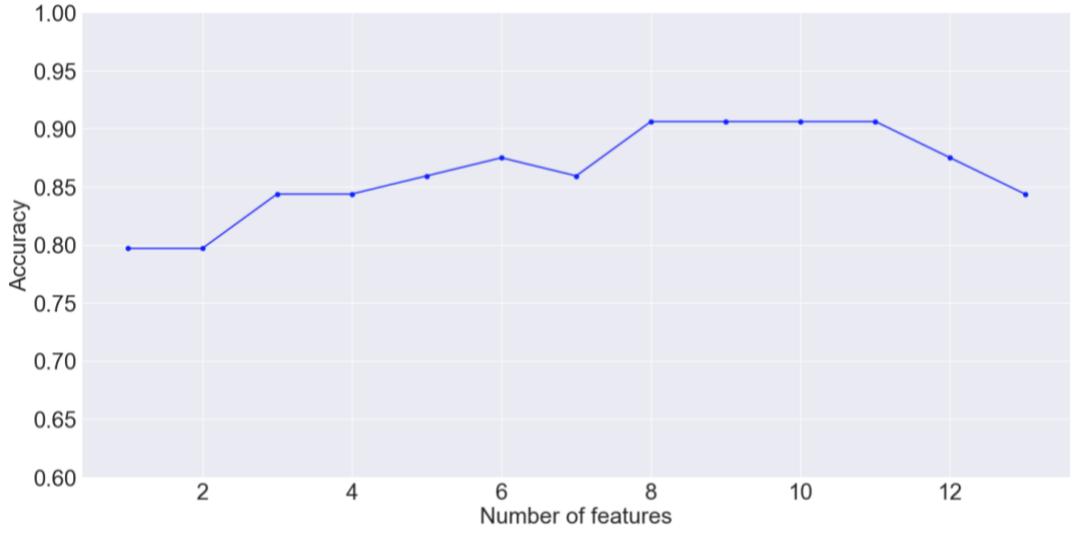
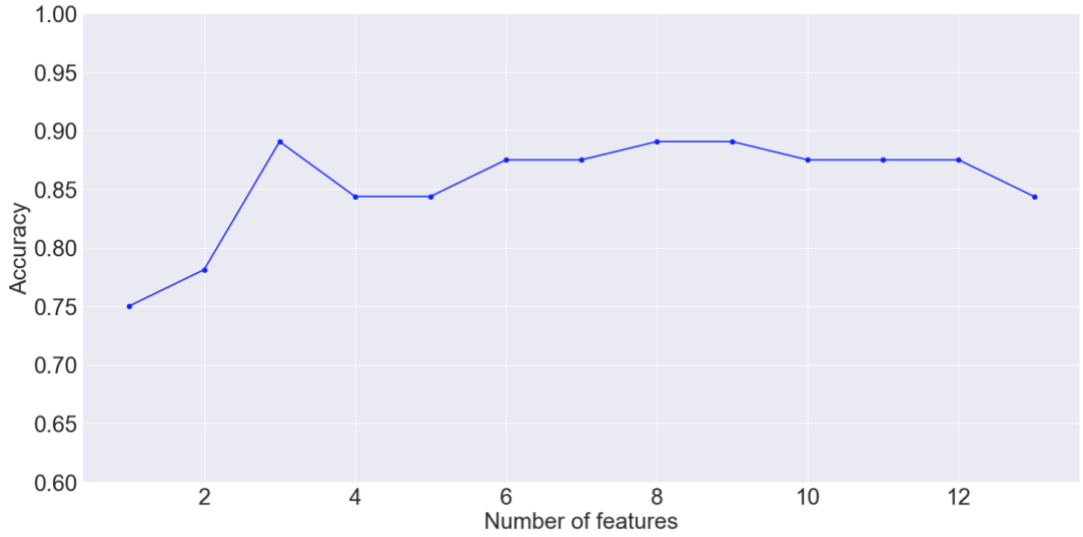


Figure 7 – Results of the sequential backwards selection using normalised feature set



On review of the results from **Figure 5-7** and as part of achieving the highest accuracy with as minimum number of features as possible, it is decided to use 8 selected features (*age, sex, cp, trestbps* and *thalach)* generated during the sequential backward selection of the standardised feature set as part training our final models.

Feature extraction

Further to selecting a reduced number of features, the process of dimensionality reduction using principle component analysis (PCA) will help to create a new feature subspace, that better models the data to account for bias and variance within the dataset.

From **Figure 8**, we understand that without standardising or normalising and having applied our PCA - the first 4 features accounts (*age, sex, cp, trestbps*) for over 98% of the variance in the dataset. Once standardised - the first 4 features no longer dominate in explaining the variance. Each features becomes more equally able to contribute to the variance of the data (**Figure 9**).

Using our training and testing dataset and an un-tuned Gaussian naïve bayes model we are able to identify that the highest accuracy for the model is achieved for at 2 principle components as shown in **Figure 10**.

Figure 8 – Explained variance ratio graph for each feature without standardising, normalising

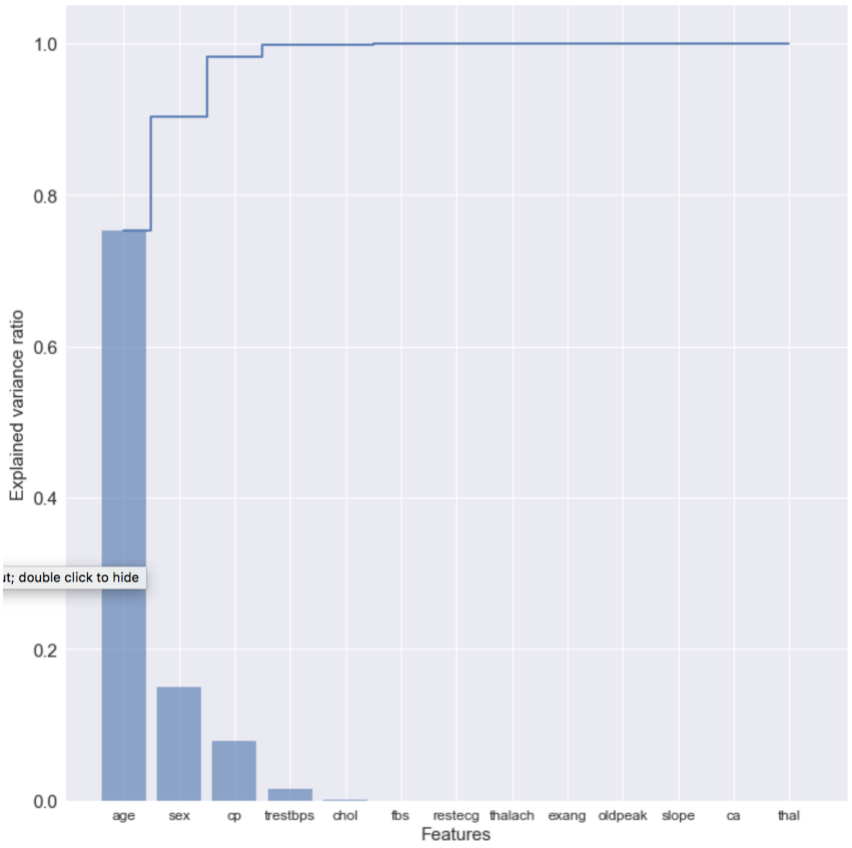


Figure 9 – Explained variance ratio graph for each PCA including standardisation of the feature set

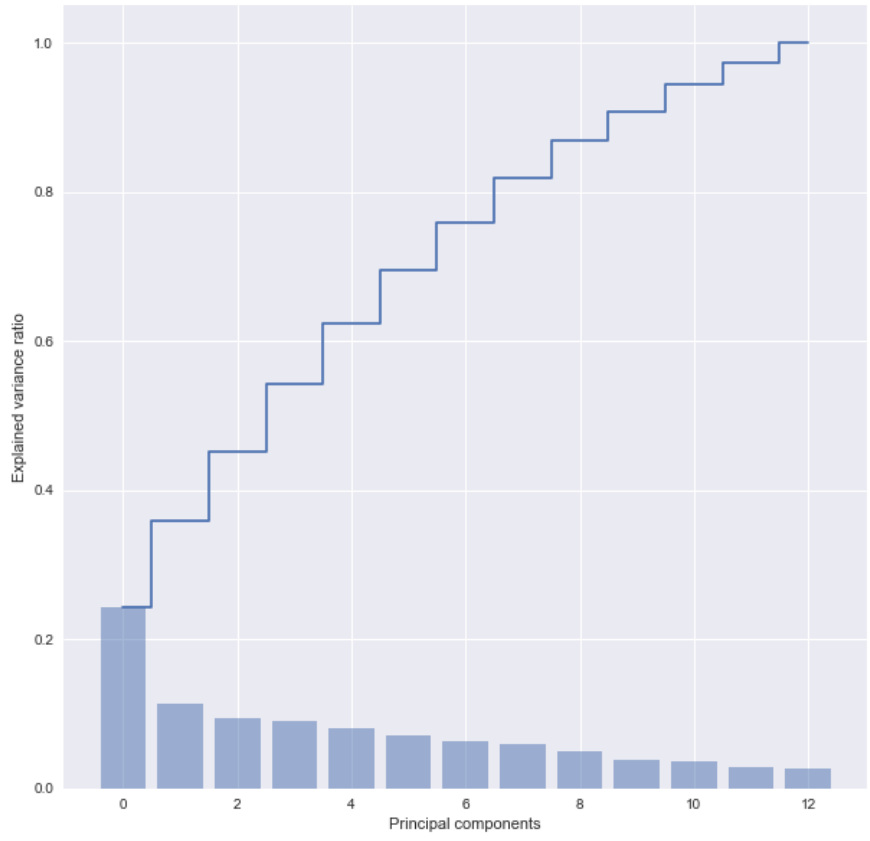
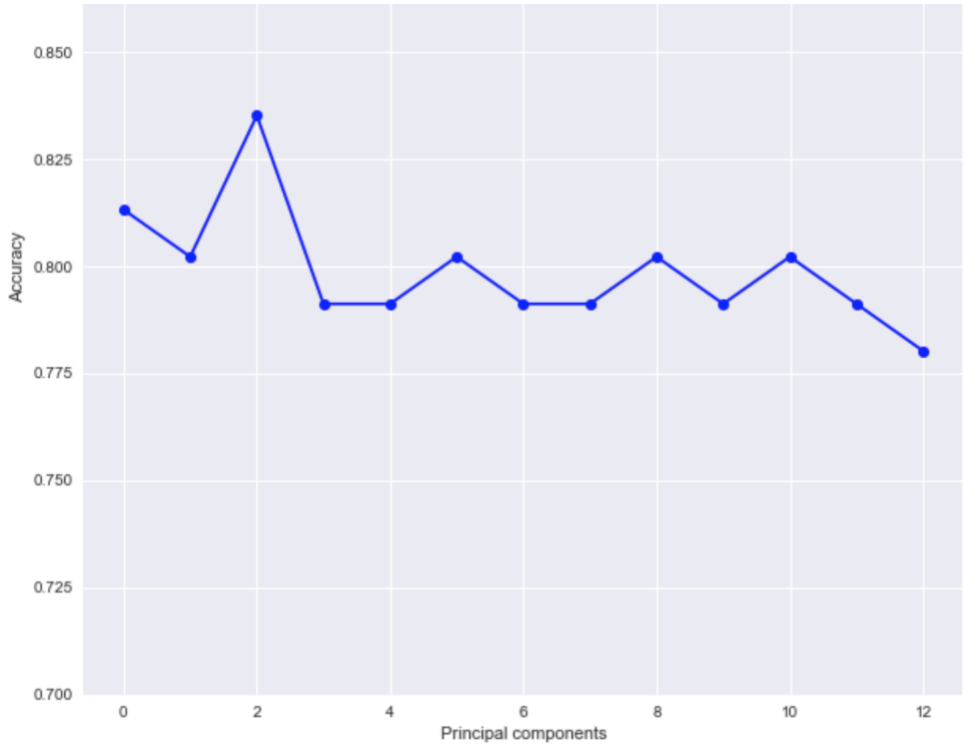


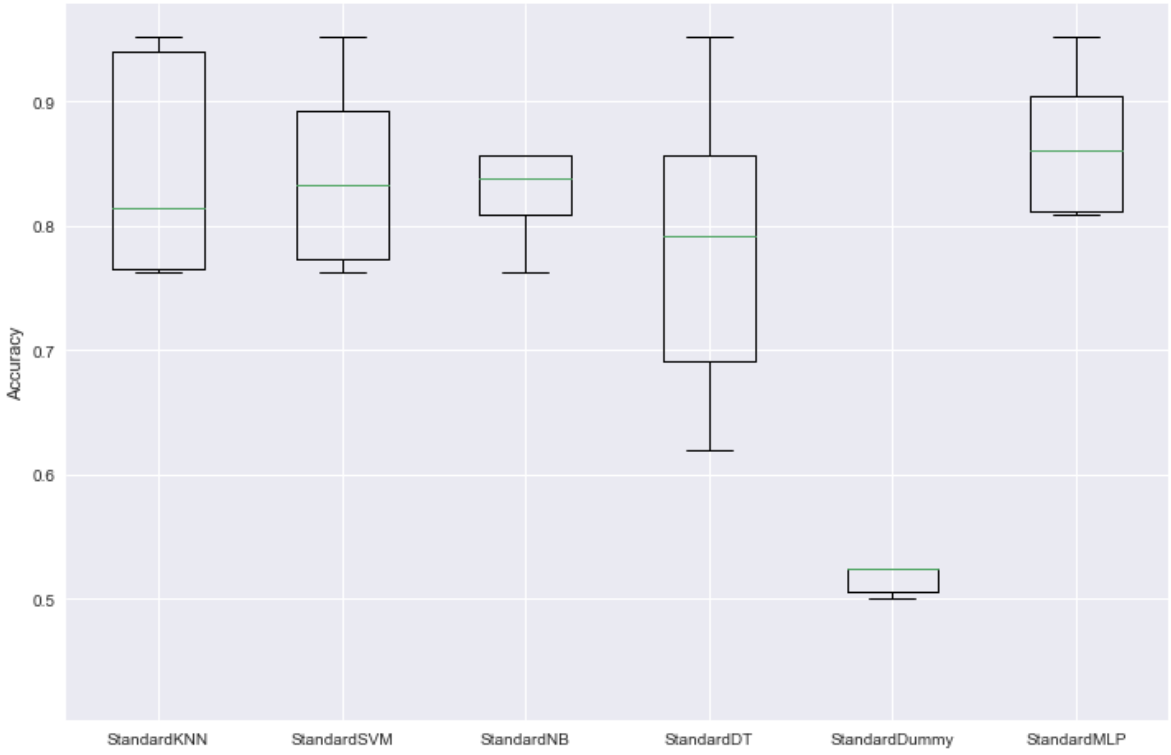
Figure 10 – Accuracy plot for each increase in the number of principle components



Model selection

In order to appropriately select two classifiers with which to train, the following untuned classifiers pictured in **Figure 11** were compared for their level of accuracy. When compared using a standard feature set and 2 principle components, it is was decided that we would select kNN and SVM classifiers for further optimisation and modelling. They both performed well and had the potential to reach the highest levels of accuracy for our dataset.

Figure 11 – Classifier algorithm comparison using standardised features with 2 principle components



Model training

Both the kNN and SVM models are implemented using their respective Scikit-learn classes.

The k-Nearest neighbours (kNN)

The principle of the kNN algorithm is to search through the training dataset for the k-most similar set of data instances. When predicting, the label for the data instance most similar to the new instance (measured via varying distance measures e.g. Euclidean and Hamming, specific to each model) is summarized and retained as the prediction for the unseen instance. Such behaviour is further explained using the pseudo code pictured in **Figure 12**.

Figure 12 – Pseudo code of the kNN classification (Tay et al. 2014)

k-Nearest Neighbor
Classify $(\mathbf{X}, \mathbf{Y}, x)$ // $\mathbf{X}$: training data, $\mathbf{Y}$: class labels of $\mathbf{X}$, x: unknown sample
for $i = 1$ **to** m **do**
  Compute distance $d(\mathbf{X}_i, x)$
end for
Compute set I containing indices for the k smallest distances $d(\mathbf{X}_i, x)$.
return majority label for $\{\mathbf{Y}_i$ where $i \in I\}$

The Support vector machine (SVM)

SVM operates by way of discriminative classification, constructing an optimal hyperplane for a multiple dimension feature space. The feature vectors belonging to the training data are used to construct the optimal hyperplane by dividing the features based on the class labels. Instances as part of the testing dataset are then assigned a class based on their geometric position relative to the minimised classifier function hyperplane. The pseudo-code explaining such behaviour is pictured in **Figure 13**.

Figure 13 – Pseudo code of the SVM classification (Pedersen & Schoeberl 2006)

Require: $\boldsymbol{X}$ and $\boldsymbol{y}$ loaded with training labeled data, $\boldsymbol{\alpha} \Leftarrow 0$ or $\boldsymbol{\alpha} \Leftarrow$ partially trained SVM

1: $C \Leftarrow$ some value (10 for example)
2: **repeat**
3: **for all** $\{\boldsymbol{x_i}, y_i\}, \{\boldsymbol{x_j}, y_j\}$ **do**
4: Optimize α_i and α_j
5: **end for**
6: **until** no changes in $\boldsymbol{\alpha}$ or other resource constraint criteria met

Ensure: Retain only the support vectors ($\boldsymbol{\alpha_i} > 0$)

Parameter and hyper-parameter optimization

Using grid search - an automated hyper-parameter tuning function (implemented using scikit-learn's *GridSearchCV* class), hyper-parameter optimization was achieved by testing for all combinations of parameters and selecting the best parameters for our model taking into account a standardised feature set and use of two principle components. The best hyper-parameters that were generated for our complete and selected feature set can be seen in **Table 3.**

Table 3 – Optimised hyper parameters for kNN and SVM with varying parameters

Classifier	Trained using all 13 features	Trained using our selected 8 features	Number of principle components	Best hyper-parameters (achieved using grid search)
kNN	**Yes**	No	2	'n_neighbors': 11
	Yes	No	4	'n_neighbors': 9
	No	**Yes**	2	'n_neighbors': 17
	No	**Yes**	4	'n_neighbors': 9
SVM	**Yes**	No	2	'C': 1.0, 'kernel': 'linear'
	Yes	No	4	'C': 1.0, 'kernel': 'linear'
	No	**Yes**	2	'C': 10.0, 'gamma': 0.01, 'kernel': 'rbf'
	No	**Yes**	4	'C': 0.01, 'kernel': 'linear'

Model evaluation

Using learning curves, comparing the training and validation accuracy of our models, helps to understand the means by which our models are learning using our training dataset. Stratified 10-fold cross-validation estimates were used to generate predictions for evaluation. A confusion matrix (**figures 16 and 17**) will also be generated for each model to identify the classification metrics defined further in **table 4.** Finally an receiver operator characteristic (ROC) plot comparing the true positive and false positive rate for both models, will be generated.

To further externally validate our model, The Tree-Based Pipeline Optimization Tool (tPOT) - an open source python automated machine learning tool that optimizes machine learning pipelines (Olson et al. 2016); will be used to independently select a model that appropriately fits our training data to achieve the highest accuracy. Using this we will further access the extent to which our model has been optimised.

Table 4 – Metrics of evaluation

Statistic	**Equation**
Sensitivity (Recall)	TP / (TP + FN)
Specificity	TN / (TN + FP)
Positive Predictive Values (Precision)	TP / (TP + FP
Negative Predictive Value	TN / (TN + FN)
F1	(2 * Precision * Recall) / (Precision + Recall)
Accuracy	(TP + TN)/ Total

3) Results

Optimising pipeline and hyper-parameters

All the results of the pipelines presented (**Table 5**) include the standard scaler and utilise the optimised hyper parameters as selected via grid search.

Table 5 – Accuracy scores using optimised hyper parameters for kNN and SVM with varying parameters

Classifier	Trained using all 13 features	Trained using our selected 8 features	Number of principle components	Accuracy score
kNN	**Yes**	No	2	0.85377
	Yes	No	4	0.84434
	No	**Yes**	2	0.85377
	No	**Yes**	4	0.84434
SVM	**Yes**	No	2	0.86321
	Yes	No	4	0.85377
	No	**Yes**	2	0.86321
	No	**Yes**	4	0.84434

From our results it suggest our kNN performed best under the utilisation of two PCAs regardless of whether it was trained on all or only our 8 selected features. For our SVM model it performed best also under the utilisation of two principle components and regardless of whether it was trained on all or only our 8 selected features. Overall the accuracy of the best SVM model (0.863) was greater than the overall accuracy of the best kNN model (0.854).

Learning curves

Using the best result parameters for each pipeline the below learning curves pictures in **Figure 14** and **15** help to diagnosis the state of learning and observe the way in which each model behaviours. **Figure 14** shows that kNN overall has a poor attitude towards learning and converges very early in the model and therefore of low variance. From **Figure 15**, we see that the SVM behaviours closer to what is expected of an appropriately learning model with a better balance between bias and variance. Achieving the best balance with 2 components and our 8 selected features.

Figure 14 - Learning curves for kNN

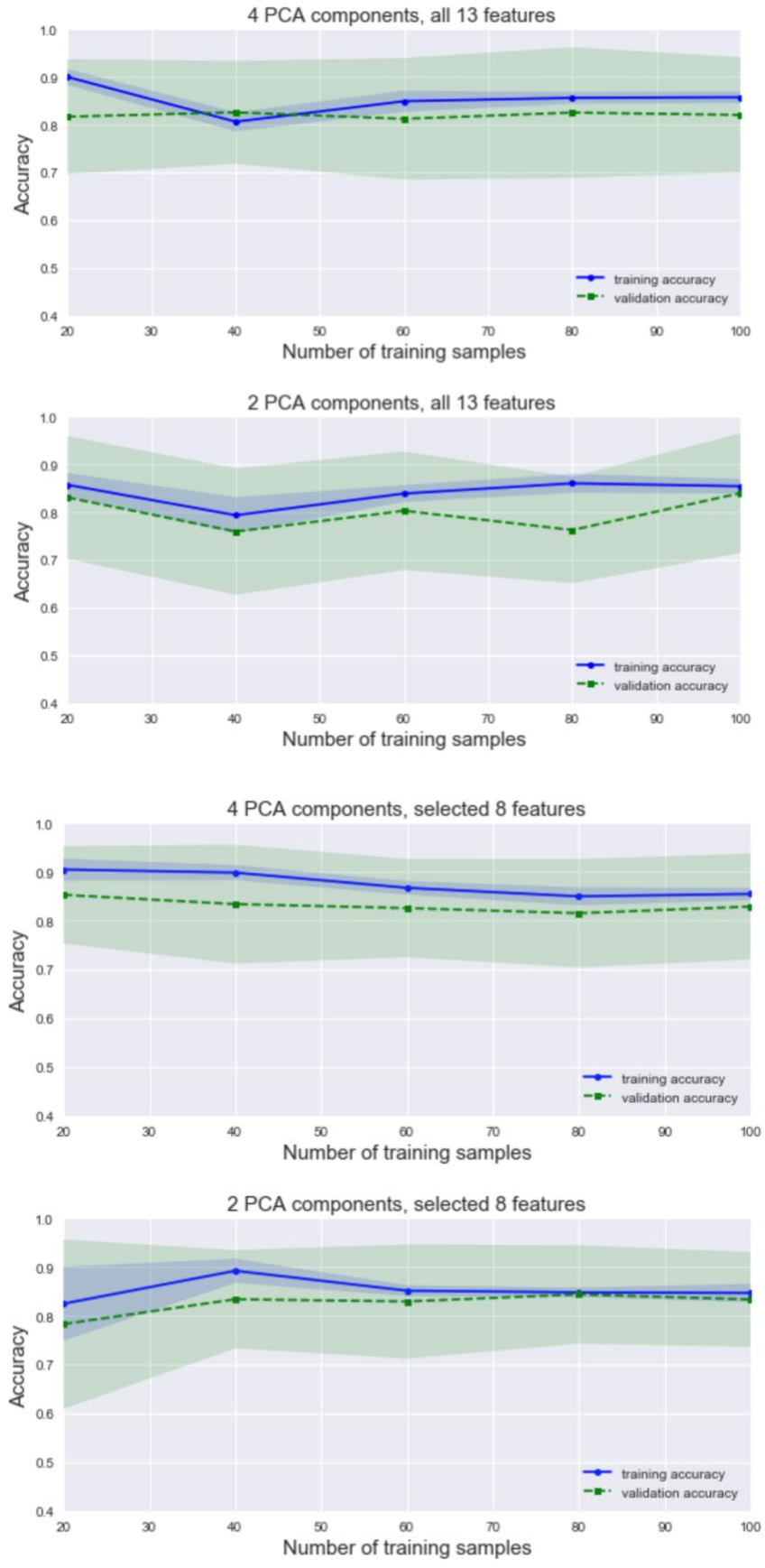
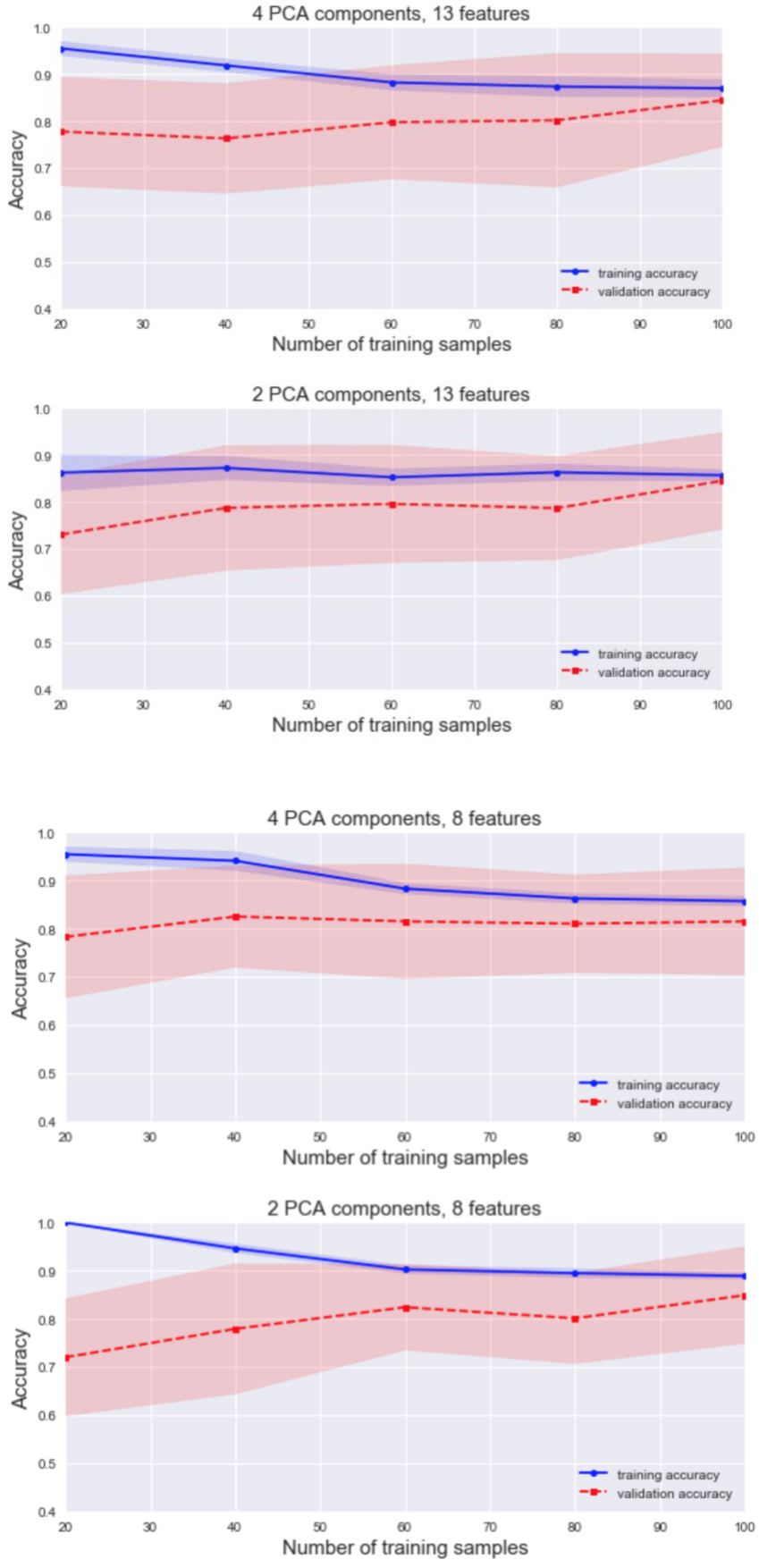


Figure 15 - Learning curves for SVM



Classification metrics

Using the 8 selected features (*age, sex, cp, trestbps* and *thalach*), standard scaler and 2 principle components the below performance metrics (**Table 6**) and confusion matrices (**Figure 16 and 17**) are from our final models – the hyper parameters for which were optimised used grid search.

Table 6 – Classification metrics using the confusion matrix for final optimised model for kNN and SVM

Classification	Sensitivity (recall score)	Specificity	PPV (precision)	NPV	F1 score	ROC AUC	Accuracy
kNN	0.827	0.769	0.827	0.769	0.827	0.798	0.802
SVM	0.818	0.806	0.865	0.744	0.841	0.804	0.813

Figure 16 – Confusion matrix for the final optimised kNN model

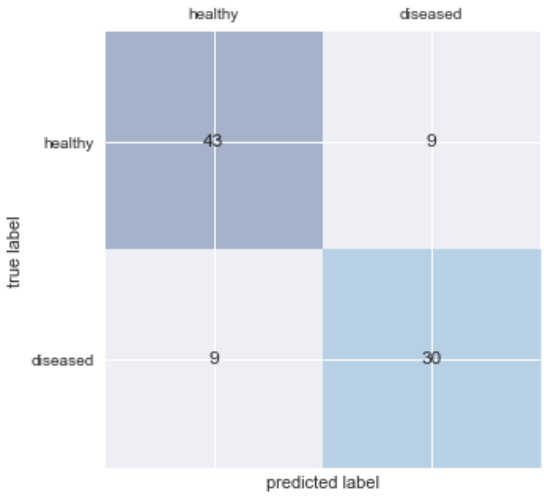
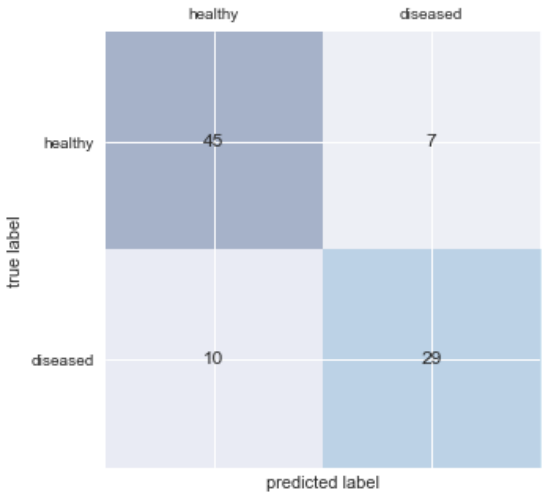


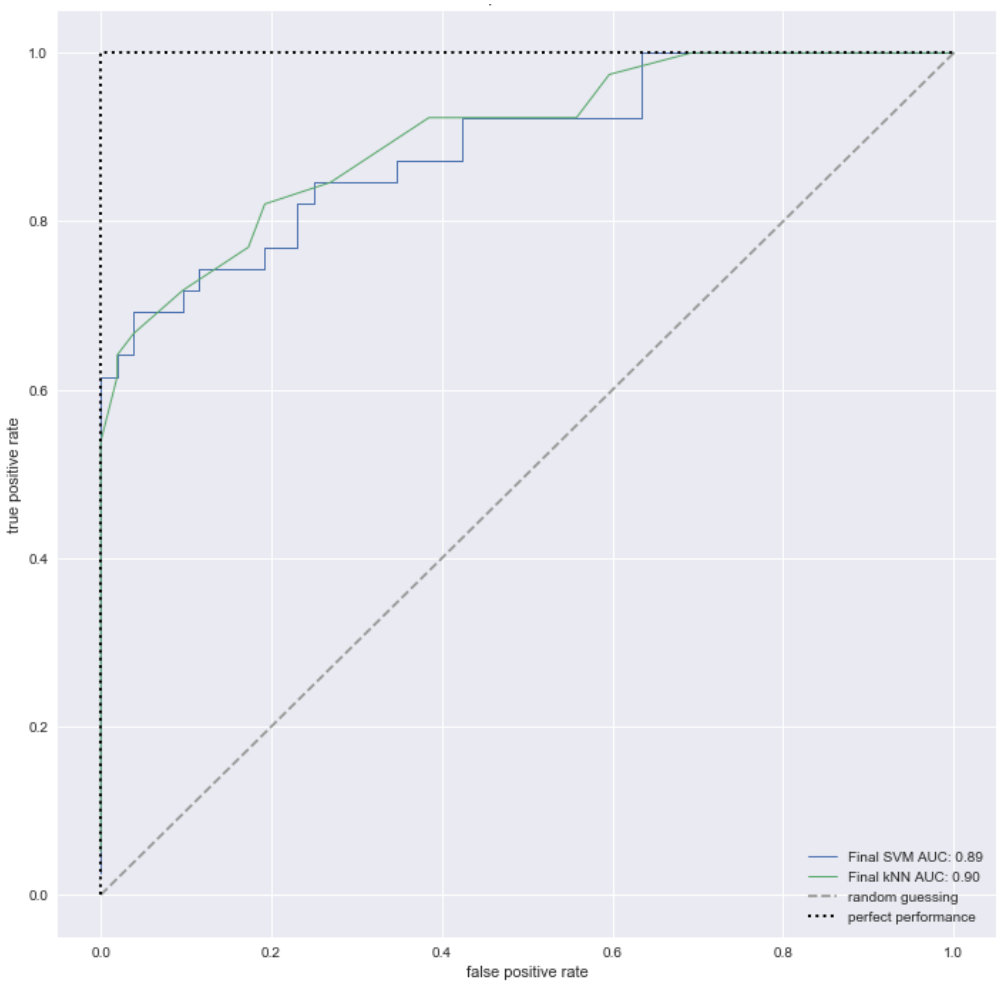
Figure 17 – Confusion matrix for the final optimised SVM model



ROC curves

ROC curves for our optimised models (**Figure 18**) are very similar and show they are have both been adequately optimised to achieve an accuracy closer to 0.8. Therefore both models are more than 80% able to correctly classify an individual with CAD from our dataset.

Figure 18 – ROC graph for our final optimised SVM and kNN model



4) Discussion and conclusion

Findings compared with other scientific publications

Both final optimised models for kNN and SVM performed similarly. Overall SVM achieved a slightly higher accuracy score of (0.813) compared to (0.802) for kNN. Yet kNN achieved a slightly higher recall score (0.827) and negative predictive value (0.769) when compared to SVM (0.818 and 0.744 respectively). In comparison to the accuracy score (0.79) generated by tPOT using a logistic regression model, it is valid to suggest both our models do well to not over or under fit our data based on this metric.

Compared to a similar study by El-Bialy et al. (2015) utilising a two separate decision tree models to predict CAD, our model performed better on the metric of accuracy. The fast decision tree and C4.5 decision tree evaluated achieved a 0.776 and 0.785 accuracy score respectively. In addition, similar to our model, both decision tree algorithm, performed comparably the same under a selected set of features, with accuracy scores of 0.781 and 0.775 respectively. Such decision tree results are further corroborated by accuracy scores generated by Chaki et al. (2015) who compares the use of SVM (0.841), C4.5 decision tree (0.776) and Naïve Bayes (0.835) model. Although higher in accuracy compared to our SVM model, the model likely included all 13 features, and failed to optimise hyper-parameters. Little evidence was provided to understand the important balance between bias and variance as our leaning curves for SVM help to explain (**Figure 15**).

Improving methodology

Although there was a reasonable balance between the instances of healthy and diseased classes (at 164 and 139 instances respectively), further techniques of over or under sampling could have been utilised to further optimise our model. Equally further testing a normalised feature set as part of our designing our final optimised model may have been justified, especially as standardising our data alters the natural distribution of our features. Finally extending our imputation method for missing values using mean imputation could help to justify median imputation as a valid technique for dealing with missing data found within our dataset.

Potential usage of trained algorithms in healthcare and health service delivery

If designed and trained to be accurate and valid enough for clinical use, such an algorithm tool to diagnosis CAD could greatly increase the number of individuals diagnosed earlier and accurately for CAD. Better early detection is likely to encourage earlier treatment and management of the condition and led to overall better outcomes for the people at greatest risk of the disease. Health service delivery in turn could be configured to generate potentially better decisions early to promote better outcomes later.

5) Appendix

References

Chaki, D., And, A. Das & Zaber, M.I., 2015. A comparison of three discrete methods for classification of heart disease data. *Bangladesh J. Sci. Ind. Res.*, 4(50), pp.293–296.

Detrano, R. et al., 1989. International application of a new probability algorithm for the diagnosis of coronary artery disease. *The American journal of cardiology*, 64(5), pp.304–10. Available at: http://www.ncbi.nlm.nih.gov/pubmed/2756873.

El-Bialy, R. et al., 2015. Feature Analysis of Coronary Artery Heart Disease Data Sets. *Procedia Computer Science*, 65, pp.459–468. Available at: http://linkinghub.elsevier.com/retrieve/pii/S1877050915029622.

Lim, M.J. & White, C.J., 2013. Coronary angiography is the gold standard for patients with significant left ventricular dysfunction. *Progress in cardiovascular diseases*, 55(5), pp.504–8. Available at: http://www.ncbi.nlm.nih.gov/pubmed/23518380.

Van Mieghem, C.A.G., 2017. CT as gatekeeper of invasive coronary angiography in patients with suspected CAD. *Cardiovascular diagnosis and therapy*, 7(2), pp.189–195. Available at: http://www.ncbi.nlm.nih.gov/pubmed/28540213.

Olson, R.S. et al., 2016. Evaluation of a Tree-based Pipeline Optimization Tool for Automating Data Science. *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference - GECCO '16*.

Pedersen, R. & Schoeberl, M., 2006. An Embedded Support Vector Machine. *In Proceedings of the Fourth Workshop on Intelligent Solutions in Embedded Systems (WISES 2006)*, pp.79–89.

Sanchis-Gomar, F. et al., 2016. Epidemiology of coronary heart disease and acute coronary syndrome. *Annals of translational medicine*, 4(13), p.256. Available at: http://www.ncbi.nlm.nih.gov/pubmed/27500157.

Tay, B., Hyun, J.K. & Oh, S., 2014. A Machine Learning Approach for Specification of Spinal Cord Injuries Using Fractional Anisotropy Values Obtained from Diffusion Tensor Images. *Computational and Mathematical Methods in Medicine*, 2014, pp.1–8. Available at: http://www.hindawi.com/journals/cmmm/2014/276589/.

Wiens, J. & Shenoy, E.S., 2018. Machine Learning for Healthcare: On the Verge of a Major Shift in Healthcare Epidemiology. *Clinical infectious diseases : an official publication of the Infectious Diseases Society of America*, 66(1), pp.149–153. Available at: http://www.ncbi.nlm.nih.gov/pubmed/29020316.