University of Cape Town

STA5069Z

Multivariate Analysis

Dimension Reduction for Prostate Cancer Prediction

Author:
Jared Lakhani

Student Number:
LKHJAR001

March 22, 2024

Project Repository

Access the source code and project files for this report on GitHub:
https://github.com/LKHJAR001/STA5069Z-Final-.git

Contents

1 Introduction

According to the World Health Organization, cancer is one of the leading causes of death, where there were nearly ten million deaths in 2020 (or one in every six deaths) [30], with prostate cancer being the most common cancer in South African men [5]. Suffice to say, predicting cancer in its early stages significantly improves the chances of successful treatment and recovery. Therefore, there's a pressing need for reliable cancer prediction tools capable of accurately classifying tumors as either malignant (harmful) or benign (non-harmful) - where to aid this process, classification methods would be utilised.

Now, most medical data sets are comprised of a plethora of features and observations - resulting in the predicting capabilities of machine learning algorithms being hampered [26]. Furthermore, a large feature set makes it difficult to visualize the data, as well as determining which feature has an impact on classification [28]. Redundant features, or ones that are derived from poor quality input would also impede the model's predictive quality [4]. Thus, the use of all the collective features would result in the curse of dimensionality - computational complexity, over-fitting and a hindering of classification performance [13]. Being such, the need for dimensionality reduction arises.

This study simply aims to ask whether there is an ideal dimension reduction technique for building tumor prediction models, and furthermore if there are ideal parameters needed to be used for each technique. Additionally, assessments will be made on the classification algorithms employed. We posit that model building on the entire feature set (without dimension reduction) will result in superior predictive quality - and propose that model building on a reduced feature set results in an adequate predictive performance.

2 Literature Review

The malignant tumor (or cancer) is one of the the leading causes of death globally [12]. Now in developing countries, seeing as there is a scarcity of well trained doctors, the inability to produce a good tumor prognosis is exacerbated. Being such, studies which sort after finding methods to predict cancer in the early stages is highly fruitful.

An extensive amount of literature using machine learning algorithms to classify whether a tumor is malignant or benign can be found. One of the more popular tumor prognosis datasets is the Wisconsin Breast Cancer dataset available in the UCI Machine Learning Repository. In [27] the classification of accuracies of four different classifiers were compared, where it was concluded that SVM achieved the highest diagnostic accuracies. Another study [7], evaluated the Area Under Curve (AUC) of three algorithms, Extreme Gradient Boost (XGBoost), Multi-layer Perceptron (MLP) and SVM with RBF kernel - where it was found that SVM achieved the best result with AUC value of 99.23%. [12] utilised dimension reduction prior to using classification using SVM-RBF kernel and XGBoost, with principal component analysis (PCA) and K-Means clustering being the dimension reduction methods. [12] concluded that K-Means used as a dimension reduction technique performs well relative to the popular PCA. Similarly, [11] utilised PCA and correlation analysis to reduce the number of dimensions prior to classification using Naive Bayes, SVM, Decision Tree, K-Nearest Neighbours, Random Decision Forest and Simple Logistic Regression. Additionally, [1] examined the performance of the same algorithms (without dimension reduction) in terms of accuracy, precision, sensitivity and specificity - to conclude that SVM had the greatest accuracy and lowest error rate of a tumor prognosis. Numerous other studies of a similar nature utilising the WBC dataset exist.

With regards to other types of tumors, [6] provides an in-depth analysis of multiple cancers, including the WBC dataset, mesothelioma, cervical cancer and prostate cancer. The study proposed a stacking-based multi-neural ensemble learning method and attained the highest level of prediction accuracy across all types of cancer data sets. Our study uses the very same prostate cancer dataset from the aforementioned study, also found in [23].

3 Data Description

The study aims to assess predictive quality of the built models on prostate tumors - where the proposed prostate tumor data set can be obtained from the Kaggle data vault in [23]. This data set was also utilised by [6] - a study which investigated deep learning techniques used in all types of cancer research. The features are calculated from a digitized image of a fine needle aspiration of a prostate tumor mass - more specifically, the cell nuclei present in the image [6]. In this case, the instances/observations are digitized images of tumors (both benign and malignant) - although it is uncertain through which means these images where obtained (how and from which organization). The data consists of 100 instances with 8 attributes/features namely Radius, Texture, Perimeter, Area, Smoothness, Compactness, Symmetry, Fractal Dimension - to be used as predictor variables for model building. And a binary response variable, namely Diagnosis Result ('0' for benign and '1' for malignant). There are no NA values present in the data, and 'malignant' tumors are the dominant class (62%). We posit that scaling the data is necessary as the area and perimeter features seem to dominate the dataset with their prominently high values relative to the other features. We also note that the aforementioned features, area and perimeter, are highly correlated (98%), and one could argue to omit one of these features - though our study retains both.

4 Methodology

4.1 Dimension Reduction

There are two approaches to dimension reduction: the first being feature selection where only certain features are selected (and the rest disgarded), and the second being compression which aims to create new features from existing ones. Moreover, feature selection in the medical field may result in a loss of crucial information (if not adequately understood through exploratory data analysis [4])- hence we will only be undergoing compression.

4.1.1 Linear Principal Component Analysis

The technique of PCA forms the bedrock for dimensionality reduction.Invented by Karl Pearson in 1901, PCA aims to find a new coordinate system such that each new dimension is orthogonal to the next and are ranked according to the amount of variation explained [2] - which has been found useful in noise reduction and retaining important information. The principal components are actually the eigenvectors of the covariance matrix of the scaled data (or correlation matrix) - where the eigenvalues represent the variation explained of the corresponding eigenvector.

4.1.2 Kernel Principal Component Analysis

Kernel principal component analysis (KPCA) is a nonlinear dimensionality reduction technique that extends traditional PCA to capture nonlinear relationships in data [17]. While traditional PCA assumes linear relationships between variables, kernel PCA allows for more flexible modeling by implicitly mapping the input data into a high-dimensional feature space using a nonlinear mapping function, known as the kernel function [9]. The key idea behind kernel PCA is to project the input data into a higher-dimensional space, where nonlinear relationships become linear or easier to separate. This is achieved by computing pairwise similarity measures, or kernel functions, between data points in the original space. Commonly used kernel functions include the radial basis function (RBF) kernel, polynomial kernel, and sigmoid kernel. Once the data is mapped into the high-dimensional feature space, PCA is performed to find the principal components, which are the directions of maximum variance in the transformed space [24].

To expound upon what the RBF and polynomial kernel is, which will be the choice of kernel in our study, let $\boldsymbol{x}, \boldsymbol{y}$ of size n be vectors in the input space. Now the RBF kernel is given as $\boldsymbol{K(x, y)} = \exp\left(\frac{-||\boldsymbol{x}-\boldsymbol{y}||^2}{2\sigma^2}\right)$ where $||\boldsymbol{x} - \boldsymbol{y}||^2$ is the squared Euclidean distance and σ is a free parameter (the width of the Gaussian distribution). And for a d degree polynomial kernel, $\boldsymbol{K(x, y)} = (\boldsymbol{x}^T\boldsymbol{y}+c)^d$, where $c \geq 0$ is a free parameter.

The study will utilize both of the aforementioned kernels, mostly due to their popularity and them being widely used.

4.1.3 Sparse Principal Component Analysis

Sparse principal component analysis is an extension of traditional PCA aimed at extracting sparse and interpretable representations of high-dimensional data [33]. In Sparse PCA, the principal components are sparse vectors with many zero entries, indicating that only a subset of the original features contributes significantly to each component. This sparsity enhances interpretability by identifying the most relevant features in the data, while discarding irrelevant or noisy ones [34].

Sparse PCA is typically formulated as an optimization problem, where the objective is to find the sparsest set of principal components that capture most of the variance in the data. This is achieved by adding an additional penalty term to the standard PCA objective function, such as the L1-norm penalty (Lasso penalty) or a combination of L1 and L2 penalties (Elastic Net penalty). So given an (n, p) data matrix X, SPCA aims to minimize the objective function $f(A, B) = \frac{1}{2}\|X - XBA^\top\|_F^2 + \Psi(B)$ where B is the sparse weight matrix and A is an orthonormal matrix. Ψ denotes a sparsity inducing regularizer (Lasso or elastic net).

However, the choice of penalty parameters and optimization algorithms can affect the performance and stability of Sparse PCA, requiring careful parameter tuning and validation. More specifically, with regard to the R package spca, there exits penalty term controls, namely α and β. The former represents the weight given to the L1-norm penalty (Lasso penalty) in the Sparse PCA objective function. A higher value of α results in sparser principal components, as it encourages more coefficients to be exactly zero. The former represents the weight given to the L2-norm penalty (Ridge penalty) in the Sparse PCA objective function. This penalty term helps to control the overall magnitude of the coefficients in the principal components. Our study uses the default parameters given in R: $\alpha = \beta = 0.0001$.

4.1.4 Robust Principal Component Analysis

Robust Principal Component Analysis (RPCA) is a technique designed to address the limitations of traditional PCA when dealing with datasets contaminated by outliers or corrupted by noise [10]. It extends PCA by explicitly modeling the presence of outliers and leveraging robust statistical techniques to estimate the principal components more effectively.

The key idea behind Robust PCA is to decompose the observed data matrix into two components: a low-rank component representing the underlying structure of the data and a sparse component representing the outliers or noise [10]. This decomposition is achieved by solving a convex optimization problem that seeks to minimize the rank of the low-rank component while promoting sparsity in the sparse component.

Our study utilises the R package robpca, where the robust loadings are computed using projection-pursuit techniques and the Minimum Covariance Determinant (MCD) method. The MCD estimator works by identifying a subset of the data (the Minimum Covariance Determinant), typically containing a majority of the "good" data points that are not contaminated by outliers [22]. It then computes the location and scatter estimates based on this subset, minimizing the determinant of the covariance matrix subject to the constraint that the subset contains a specified fraction of the observations.

4.1.5 Principal Curves

The concept of principal curves was introduced by Hastie and Stuetzle in 1989 as a means to uncover the "central tendency" of data points in a high-dimensional space. They defined principal curves as curves that pass through the "middle" of the data cloud, minimizing the sum of squared perpendicular distances from the data points to the curve [8]. This formulation ensures that the curve effectively captures the essential features of the dataset while discarding noise and outliers. Unlike linear techniques such as PCA, which seek linear projections that maximize variance, principal curves aim to capture the underlying structure of the data through smooth, nonlinear paths. Similarly to PCA however, the transformed points are

generated by the projections of the original points onto the principal curve (or principal component in the case of PCA) and are measured by their arc lengths.

4.1.6 Cluster Analysis

Clustering is a method used in unsupervised machine learning to identify inherent structures within data by grouping similar observations based on certain criteria. These criteria can vary depending on the context and nature of the data. Generally, similarity between observations is determined by the distance or dissimilarity between data points in a multidimensional space [31]. Other criteria include clustering based on probability, modality or density. Probability-based clustering assigns points to clusters based on the probability distribution that is most likely to have generated them. For instance, a point's value might determine which distribution it is more likely to belong to. Modality clustering identifies clusters based on the number of peaks in the density of observations. If there are multiple peaks in the density plot of a variable, it suggests the existence of multiple clusters. Density clustering identifies clusters based on regions of high density within a dataset. Points within these high-density regions are grouped together to form clusters.

Clustering can be applied to observations, variables, or both. When clustering observations, the goal is to group similar data points together based on their feature values. On the other hand, clustering variables involves grouping features that exhibit similar patterns across observations. This can be useful for identifying redundant or highly correlated variables in datasets [3]. Two-way clustering, also known as biclustering, simultaneously clusters both observations and variables. This technique is particularly beneficial when exploring datasets where the relationship between observations and variables is complex or when patterns of interest are present in subsets of both observations and variables [16]. Two-way clustering is popular in gene expression analysis in bioinformatics.

4.1.6.1 Hierarchical Clustering Hierarchical clustering is a method of unsupervised machine learning used to organize data into a hierarchical structure of clusters. Unlike other clustering methods, hierarchical clustering creates a tree-like hierarchy of clusters, known as a dendrogram, where clusters at each level of the tree are formed by merging or splitting existing clusters based on similarity [18].

There are two main types of hierarchical clustering: agglomerative and divisive. In agglomerative hierarchical clustering, each data point starts as its own cluster, and pairs of clusters are iteratively merged together based on a chosen similarity measure until all points belong to a single cluster. Divisive hierarchical clustering, on the other hand, starts with all data points in one cluster and recursively splits them into smaller clusters until each point is in its own cluster.

4.1.6.2 Non-Hierarchical or Partitioning Clustering Non-hierarchical clustering, also known as partitioning clustering, refers to a class of clustering algorithms that directly divide data points into a pre-defined number of clusters without forming a hierarchical structure. Unlike hierarchical clustering, which creates nested clusters, partitioning methods assign each data point to exactly one cluster [20].

One of the most popular partitioning clustering algorithms is K-means clustering. In K-means, the algorithm aims to partition the data into K clusters, where K is specified by the user. The algorithm iteratively assigns each data point to the nearest cluster centroid and updates the centroids based on the mean of the points assigned to each cluster. This process continues until convergence, typically when the assignments of data points to clusters no longer change significantly [14]. We aim to utilize K-means clustering, along with other partitioning methods.

Another widely used partitioning clustering algorithm is Gaussian Mixture Models (GMM). GMM represents the distribution of data as a mixture of several Gaussian distributions, each corresponding to a cluster. The algorithm estimates the parameters of these Gaussian distributions, including mean and covariance, to maximize the likelihood of observing the data [32]. Data points are then assigned to clusters

based on their probability of belonging to each Gaussian distribution.

4.2 Classification Methods

Classification methods are systematic approaches used to construct classifiers from input datasets. These classifiers are built based on a learning target function that maps each feature set to predetermined class labels [12]. The process of classification involves two main steps.

Firstly, a classification algorithm constructs the classifier by analyzing a training set composed of database tuples and their corresponding class labels. This phase, often referred to as supervised learning, entails providing the class label for each training tuple. During this step, the algorithm learns the relationships between the features and the class labels, enabling it to make accurate predictions.

In the second phase, the trained classifier is utilized for classification. Given a new set of features, the classifier assigns it to one of the predefined classes based on the learned patterns from the training data. This step allows the classifier to generalize its predictions to unseen data.

4.2.1 Support Vector Machines

Support Vector Machine (SVM) works by finding the optimal hyperplane that best separates data points into different classes while maximizing the margin between classes. SVM is effective in high-dimensional spaces and is versatile due to its ability to handle linear and nonlinear data through the use of kernel functions. It aims to find the decision boundary that maximizes the margin, making it robust to outliers [25]. We propose to use the RBF kernel in our study, purely because it has been shown to give rise to higher performance measures as expounded on in Section 2, with default parameters in the R package svm: cost $= \gamma = 1$.

4.2.2 Logistic Regression

Logistic regression is a statistical method used for binary classification tasks, where the goal is to predict the probability of a binary outcome based on one or more predictor variables [15]. Despite its name, logistic regression is a classification algorithm rather than a regression one. It models the relationship between the predictor variables and the probability of the binary outcome using the logistic function.

In logistic regression, the logistic function, also known as the sigmoid function, transforms the linear combination of predictor variables into a probability score between 0 and 1. This probability score represents the likelihood of the binary outcome belonging to a particular class. The logistic regression model is trained using optimization techniques such as maximum likelihood estimation or gradient descent, where the model parameters are adjusted to minimize the difference between predicted probabilities and actual class labels in the training data [19].

4.2.3 Naive Bayes

Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem with the "naive" assumption of feature independence. It calculates the probability of a class given a set of features by multiplying the probabilities of each feature occurring in that class and normalizing by the probability of the features occurring together [21]. It requires a small amount of training data to estimate the necessary parameters, making it computationally efficient [29]. Naive Bayes comes in several variants, including Gaussian, Multinomial, and Bernoulli, each suited to different types of data. Gaussian Naive Bayes assumes that continuous features follow a Gaussian distribution, while Multinomial Naive Bayes is designed for discrete features often encountered in text classification tasks. Bernoulli Naive Bayes is suitable for binary feature vectors.

4.3 Classifier Performance Metrics

A classifier can be defined as a mapping from data instances to predicted classes [12] - and with prostate tumor prediction, these predicted values are binary i.e. malignant or benign. Four possible outcomes exist: if the tumor is malignant and classified as malignant, it is called true positive (TP). True negative (TN) is if the tumor is benign and classified as benign. False positive (FP) is if the tumor is benign but classified wrongly as malignant. Lastly, false negative (FN) is if the tumor is actually malignant but misclassified as benign - the most fatal misclassification in tumor prognosis.

We examine accuracy $= \frac{TP+TN}{TP+FP+FN+TN}$ i.e the ratio of correctly classified to total predictions (misleading at times if the data set is unbalanced). Popular metrics in the medical field [12] are sensitivity $= \frac{TP}{TP+FN}$ and specificity $= \frac{TN}{TN+FP}$. The former being the proportion of malignant prostate tumors correctly classified by the model, and the latter being the proportion of benign prostate tumors correctly classified by the model. To ensure death avoidance, it is vital to minimize the false negatives - which would imply a high sensitivity is sought after in tumor prognosis models. However, one cannot now say that a model's sensitivity is the sole performance metric to assess. That is, a model with a high sensitivity but low accuracy and specificity would suggest it classified most tumors as malignant - an imprudent approach to modelling.

Now other performance metrics can be assessed, popular metrics for tumor prediction models include recall, precision and the F-1 score as used in [1] and [4]. Our study does not include said metrics.

4.4 Prognosis Model

A cancer prediction is a classification or prognosis of whether a given tumor is benign or malignant. Frameworks of dimension reduction techniques to be utilized for the model building are principal component analysis (including kernel, Sparse and Robust PCA), principal curves and cluster analysis. The latter is quite a peculiar means of dimension reduction - where this unsupervised machine learning technique will create clusters to serve as the new feature as used in [12]. Figure 1 aids in illustrating the prognosis model - where the specified data set (with the full feature set) is split into both a training and testing set. Each phase consists of dimension reduction which reduces the feature space, where these features are then used to generate the model in the training phase by using classification algorithms: SVM-RBF kernel, Logistic Regression and Naive Bayes. After which, the generated model is used to classify whether a given tumor is benign or malignant in the testing phase, and compared with the response variable in the test set to derive an accuracy, sensitivity and specificity.

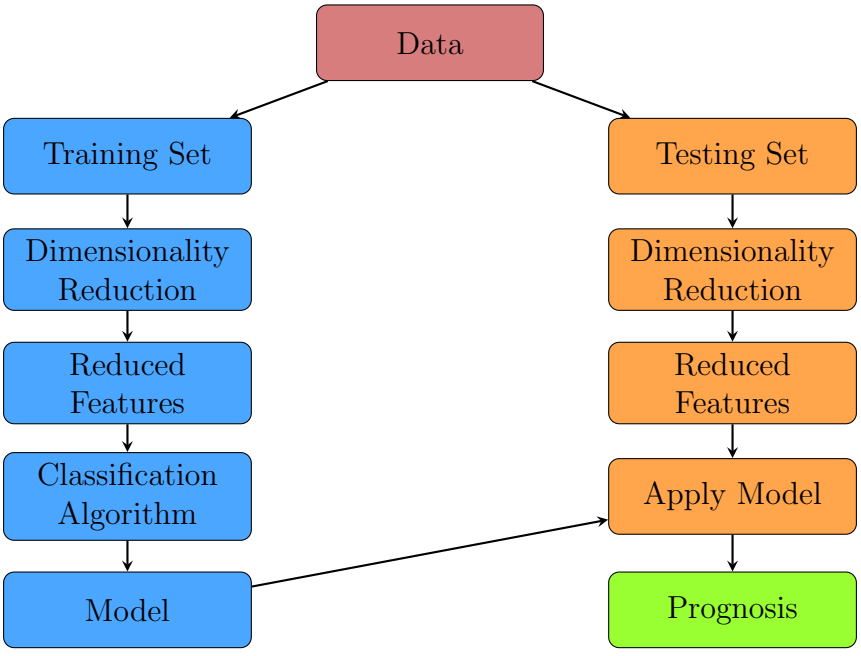


Figure 1: Flow Diagram of Prognosis Model

Moreover, with regard to splitting the data into a training and testing set, repeated cross-validation was employed due to its straight forward implementation and versatility - the high cost of computation was an issue however. The means of the trials were then used for analysis.

4.5 Visualization of Methodology

We expound upon the methodology through the use of just two examples of dimension reduction methods used in the study, namely linear PCA and cluster analysis.

4.5.1 PCA Example (Using SVM and 80/20 Train-Test Split)

We aim to shed light on the inner workings of the prognosis model by illustrating what occurs when we use PCA as the means of dimensionality reduction. PCA also serves to simplify the data by linearly altering the data and creating a new coordinate system with the largest retained variance. Figure 2 shows the first two principal coordinates after dimension reduction on the training set - we note the tumor type is not entirely separable so we plot the first three principal components (of the training data) in Figure 3.

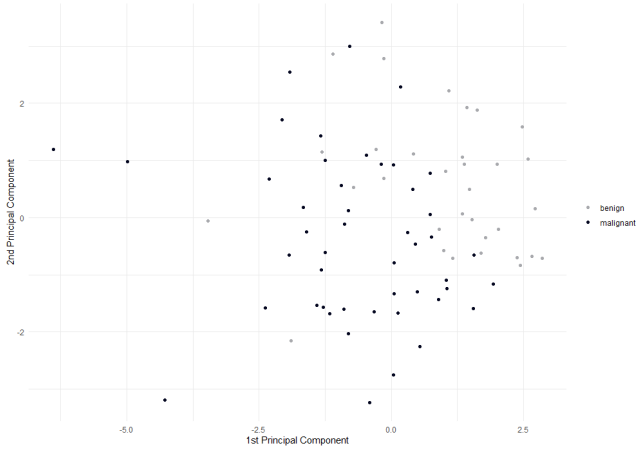


Figure 2: First 2 Principal Components for Training Data (80/20 split)

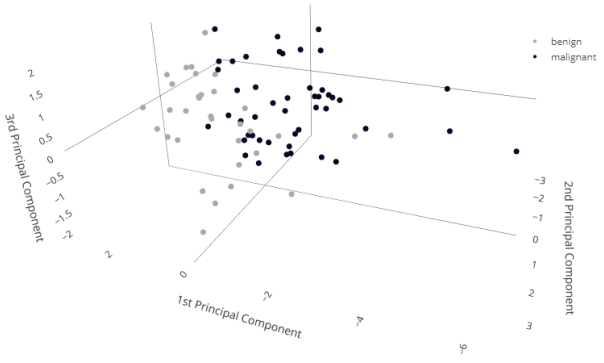


Figure 3: First 3 Principal Components for Training Data (80/20 split)

Using the first two principal components for dimension reduction in both the training phase (to build the classifier with SVM) and testing phase, we apply the model to the testing set with reduced features to make predictions. We compare these predictions to the response variable in the testing set - where each comparison is given as a true positive (TN), true negative (TN), false positive (FP) and false negative (FN) as seen in Figure 4. The same methodology applied to the first three principal components can be seen in Figure 5.

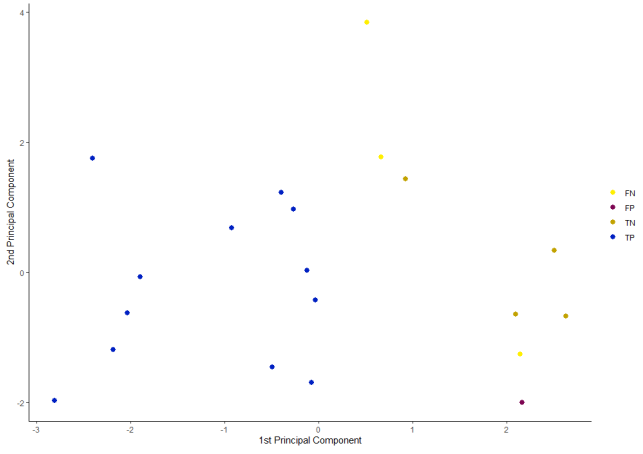


Figure 4: Model Predictions using 2 Principal Components (SVM and 80/20 split)

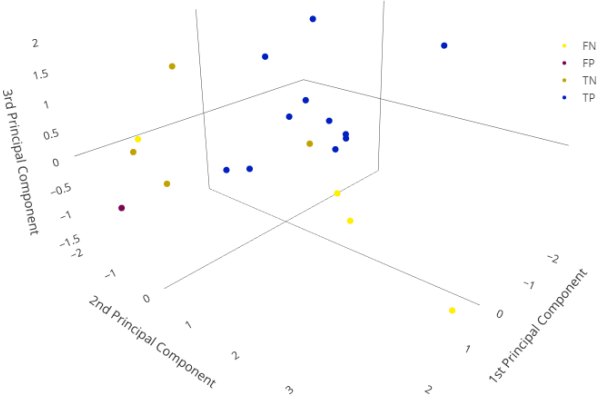


Figure 5: Model Predictions using 3 Principal Components (SVM and 80/20 split)

4.5.2 Clustering Example (Using 2 K-Means Clusters, SVM and 80/20 Train-Test Split)

Similarly, we visualize the prognosis methodology utilizing K-Means clustering as the means of dimension reduction - where the new feature is the cluster to which the particular observation belongs. In this example, observations will be grouped into 2 clusters (whereby the new feature created will be the cluster to which the observation belongs- either cluster 1 or 2). Figure 6 shows clustering undergone on the training set, where we see most of the malignant tumors have been grouped to the first cluster.

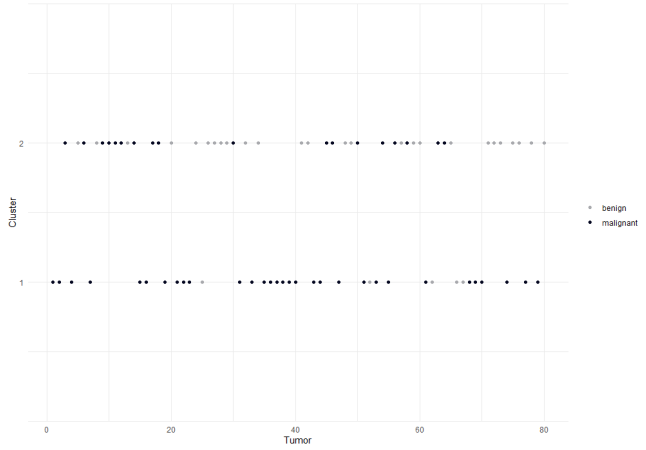


Figure 6: 2 Clusters on Training Data (SVM and 80/20 split)

Furthermore, after building the classifier using 2 clusters, we make predictions on the response variable of the testing set as seen in Figure 7.

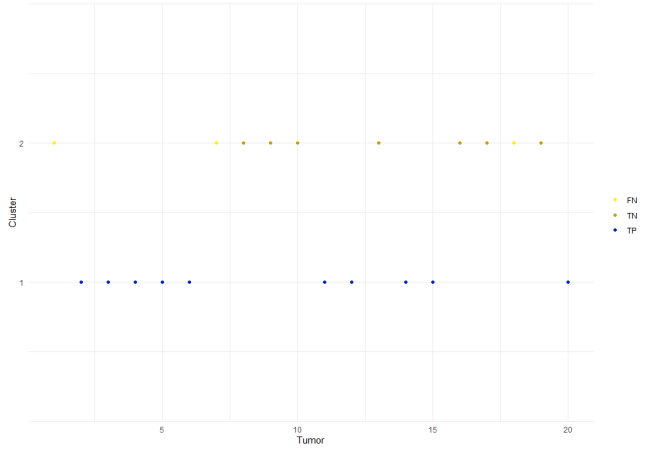


Figure 7: Model Predictions using 2 Clusters (SVM and 80/20 split)

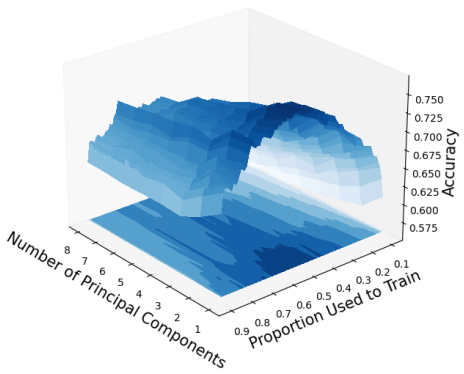
5 Analysis and Results

One of our objectives of this study was to determine if there were ideal parameters to utilize in our dimension reduction methods, namely linear, kernel, sparse and robust principal component analysis, as well as principal curves and cluster analysis. Classification methods: Support Vector Machines (RBF-kernel), Logistic Regression and Naive Bayes were to be critiqued as well. Additionally, we aimed to determine if there were any superior dimension reduction techniques and how they compared with others.

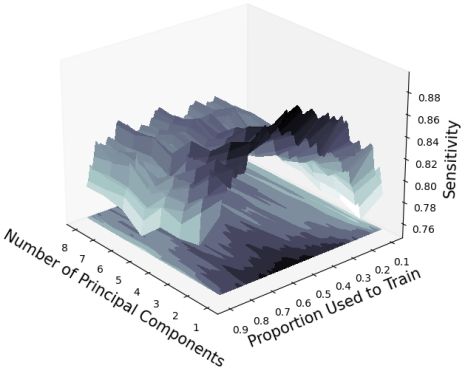
5.1 Linear Principal Component Analysis

Particularly with respect to PCA - we aimed to ascertain if there were an ideal number of PCs to be used to train the model in the training phase, as well as dimension reduction used on the testing set in the testing phase. Additionally, differences in performance measures that were brought about by varying

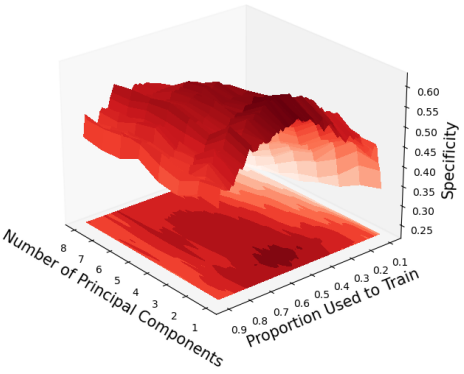
the train-test split of the data seemed worthwhile to investigate. Subsequently, Figures 8a - 10c illustrate the nature of performance measures as a function of both the number of principal components used in dimension reduction (the same number used in both the training and testing phase) and the train-test split.



(a) Accuracy vs Number of PCs and Train-Test Split

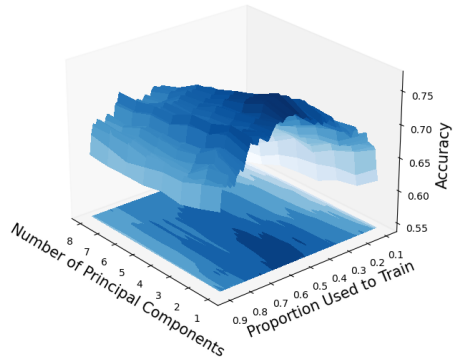


(b) Sensitivity vs Number of PCs and Train-Test Split

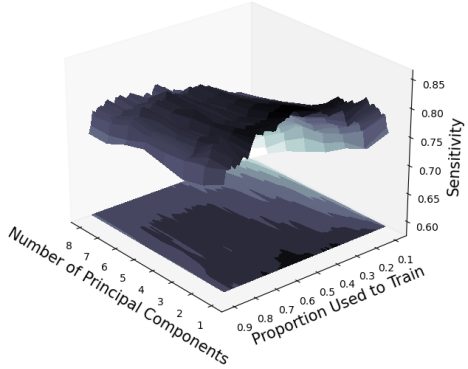


(c) Specificity vs Number of PCs and Train-Test Split

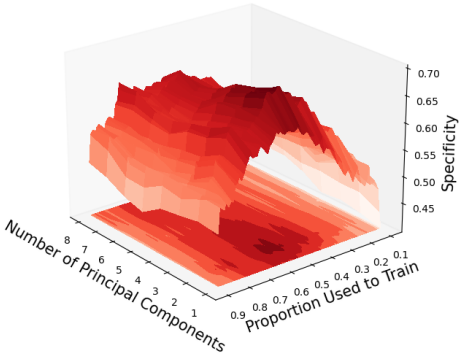
Figure 8: Performance Measures vs Number of PCs and Train-Test Split using SVM (100 Trials)



(a) Accuracy vs Number of PCs and Train-Test Split

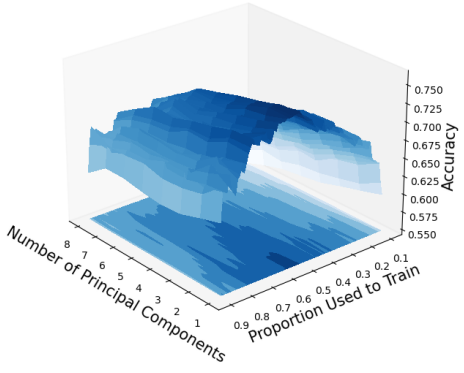


(b) Sensitivity vs Number of PCs and Train-Test Split

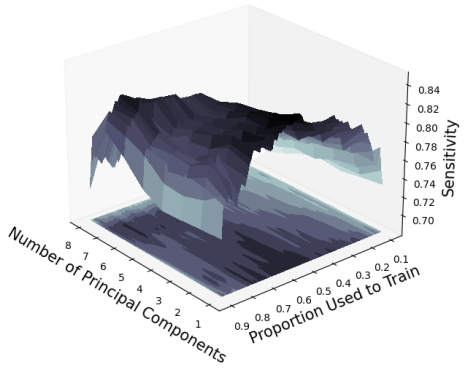


(c) Specificity vs Number of PCs and Train-Test Split

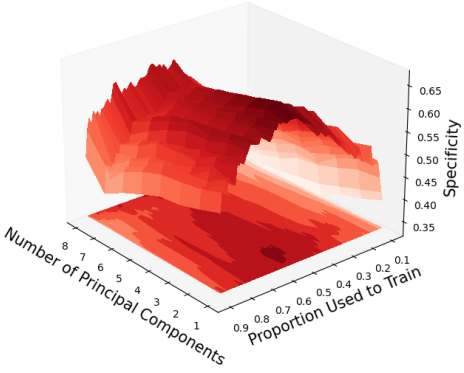
Figure 9: Performance Measures vs Number of PCs and Train-Test Split using Logistic Regression (100 Trials)



(a) Accuracy vs Number of PCs and Train-Test Split



(b) Sensitivity vs Number of PCs and Train-Test Split



(c) Specificity vs Number of PCs and Train-Test Split

Figure 10: Performance Measures vs Number of PCs and Train-Test Split using Naive Bayes (100 Trials)

It is apparent that there is a clear pattern to the nature of the performance measures - there seems to be a noticeable decrease in prediction quality of the model as the number of principal components used to

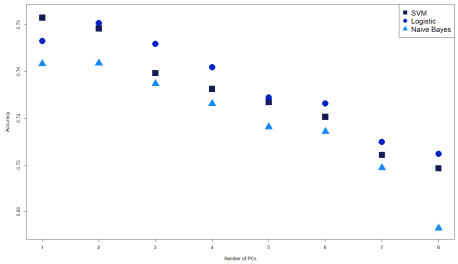
train said model, increase. Both accuracy and sensitivity peak at using a single principal component, yet specificity seems to peak at using two. Intuitively, one may think that the more principal components used, the greater the amount of total variation of the data explained, hence a greater quality model. Yet with this notion we forego the possibility that more principal components may include increasing the amount of redundant or irrelevant information in the model and furthermore, overfitting ie. the model learned to fit the noise in the data and not the underlying patterns.

We also notice that the greatest performance measures are obtained when the train-test split is approximately 55/45, being such, we fix the split and investigate further in two dimensions. We see comparisons of performance measures between the different classification methods and number of principal components used in Table 1. We note the lower bounds of the 95% confidence intervals suggesting the distributions of the performance measures are left-skewed. We also observe the larger standard deviations of the specificities.

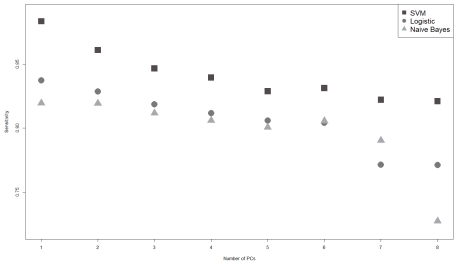
The plotted means (of 1000 trials) for each of the performance measures for each classification method is given in Figures 11a - 11c. We note the decrease in all performance measures as the number of principal components increase for all classification methods, as mentioned previously. We note the first PC captures approximately 37.5% of the total variation - hence we cannot attribute the exceptional performance of using the first PC to it explaining the majority of the variation in the data. Additionally, there seems not to be a classification method which consistently gives rise to the best prediction quality - that is, there is no single best classification method.

Table 1: Performance Measures with Varying Number of PCs and Classification Methods for 55/45 Train-Test Split (1000 Trials)

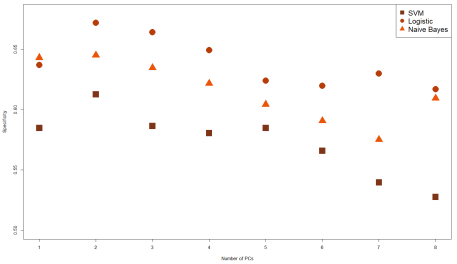
		Accuracy			Sensitivity			Specificity		
PCs	Classification	Mean	sd	95% CI	Mean	sd	95% CI	Mean	sd	95% CI
1	SVM	0.763	0.102	(0.467, 0.889)	0.884	0.095	(0.633, 1)	0.585	0.186	(0.050, 0.846)
	Logistic	0.753	0.110	(0.4, 0.889)	0.837	0.119	(0.538, 1)	0.637	0.183	(0.071, 0.882)
	Naive	0.743	0.111	(0.4, 0.889)	0.820	0.127	(0.520, 1)	0.643	0.185	(0.059, 0.882)
2	SVM	0.758	0.104	(0.466, 0.889)	0.861	0.105	(0.613, 1)	0.613	0.180	(0.133, 0.875)
	Logistic	0.761	0.110	(0.422, 0.911)	0.829	0.112	(0.567, 1)	0.672	0.181	(0.158, 0.929)
	Naive	0.744	0.109	(0.422, 0.889)	0.819	0.120	(0.555, 1)	0.645	0.186	(0.130, 0.929)
3	SVM	0.739	0.103	(0.444, 0.889)	0.847	0.104	(0.6, 1)	0.586	0.183	(0.117, 0.875)
	Logistic	0.752	0.111	(0.422, 0.889)	0.819	0.112	(0.548, 1)	0.664	0.180	(0.200, 0.929)
	Naive	0.735	0.106	(0.444, 0.889)	0.812	0.115	(0.548, 1)	0.635	0.186	(0.158, 0.923)
4	SVM	0.733	0.100	(0.466, 0.867)	0.840	0.106	(0.594, 1)	0.580	0.177	(0.143, 0.867)
	Logistic	0.742	0.109	(0.422, 0.889)	0.812	0.110	(0.560, 1)	0.649	0.181	(0.2, 0.929)
	Naive	0.726	0.104	(0.422, 0.867)	0.806	0.114	(0.548, 1)	0.622	0.186	(0.143, 0.923)
5	SVM	0.727	0.101	(0.444, 0.889)	0.829	0.110	(0.576, 1)	0.585	0.184	(0.143, 0.875)
	Logistic	0.729	0.106	(0.444, 0.867)	0.806	0.107	(0.560, 1)	0.624	0.177	(0.176, 0.923)
	Naive	0.716	0.101	(0.422, 0.867)	0.801	0.114	(0.545, 1)	0.604	0.182	(0.158, 0.9)
6	SVM	0.721	0.098	(0.444, 0.867)	0.831	0.108	(0.581, 1)	0.566	0.187	(0.117, 0.875)
	Logistic	0.726	0.104	(0.444, 0.867)	0.804	0.105	(0.567, 1)	0.620	0.176	(0.182, 0.923)
	Naive	0.714	0.102	(0.422, 0.867)	0.806	0.113	(0.548, 1)	0.591	0.184	(0.133, 0.9)
7	SVM	0.704	0.095	(0.444, 0.845)	0.822	0.108	(0.581, 1)	0.540	0.188	(0.111, 0.857)
	Logistic	0.710	0.104	(0.4, 0.867)	0.771	0.114	(0.516, 0.962)	0.630	0.173	(0.176, 0.923)
	Naive	0.699	0.101	(0.422, 0.844)	0.790	0.114	(0.531, 0.963)	0.575	0.183	(0.133, 0.875)
8	SVM	0.699	0.091	(0.467, 0.844)	0.821	0.108	(0.588, 1)	0.528	0.192	(0.095, 0.857)
	Logistic	0.705	0.101	(0.422, 0.844)	0.771	0.117	(0.515, 0.962)	0.617	0.177	(0.174, 0.923)
	Naive	0.673	0.098	(0.4, 0.822)	0.727	0.119	(0.484, 0.926)	0.609	0.179	(0.166, 0.923)



(a) Accuracy (Mean) vs Number of PCs



(b) Sensitivity (Mean) vs Number of PCs



(c) Specificity (Mean) vs Number of PCs

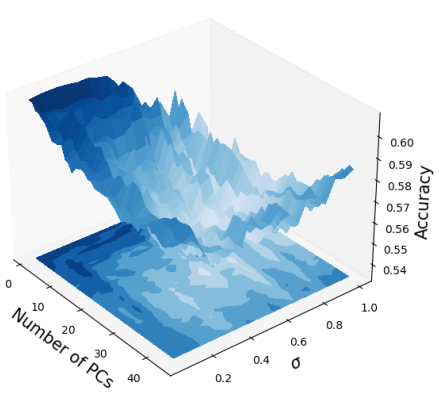
Figure 11: Performance Measures (Mean) vs Number of PCs for 55/45 Train-Test Split (1000 Trials)

5.2 Kernel Principal Component Analysis

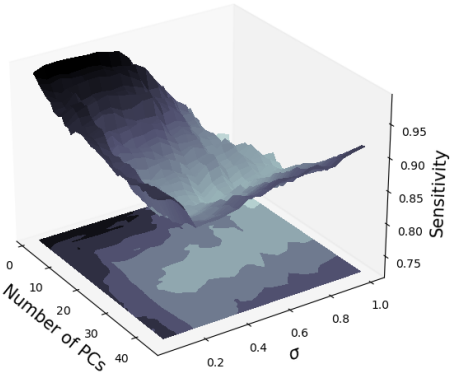
We also undergo kernel PCA - as the underlying structure of the data may be nonlinear, and kernel PCA can capture complex nonlinear relationships among the variables. With kernel PCA, as before, we aim to ascertain whether there are an ideal number of kernel principal components to be used to train the model. Unlike before however, the maximum number of principal components are not limited to the amount of features (in this case there were eight), but rather, limited to the rank of the kernel matrix. More importantly however, we aim to determine if kernel PCA gives rise to higher performance measures relative to linear PCA, or if kernel PCA is even worth considering at all. Note, we fix the train-test split to 55/45 so to ensure comparability amongst results.

5.2.1 RBF Kernel

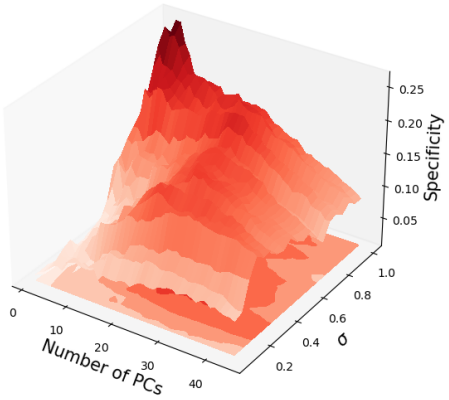
Furthermore, with respect to using a radial basis function as the kernel, one is now prompted to investigate if there is a certain σ which could potentially give rise to better model performance. Following Figures 12a - 14c, which illustrate performance measures for varying ranges of σ and number of principal components, there seems not to be an overarching common trend - apparent from the irregular shapes. For accuracy and sensitivity plots however, one can construe that there is a decreased performance when increasing the number of principal components (for SVM and logistic regression classification methods) - yet this notion is not apparent for specificity. Noticeable values of σ which bring about high accuracy and sensitivity are $\sigma = \{0.05, 0.1, 0.15, 0.5\}$. For specificity, the distinguishable values are $\sigma = \{0.3, 0.5, 0.6, 1\}$.



(a) Accuracy vs σ and Number of PCs

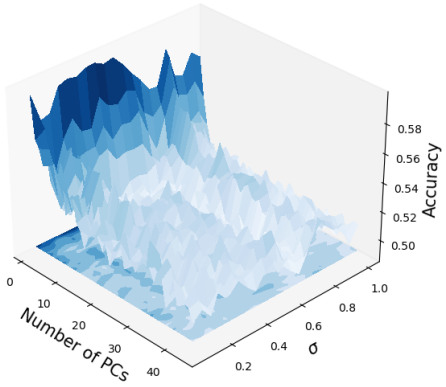


(b) Sensitivity vs σ and Number of PCs

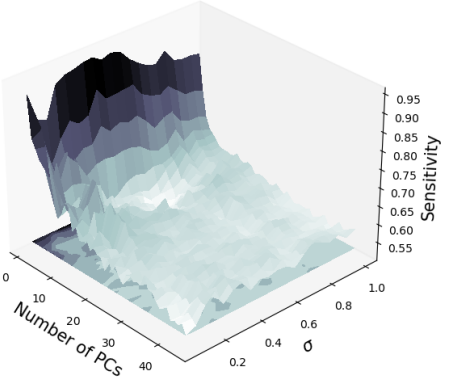


(c) Specificity vs σ and Number of PCs

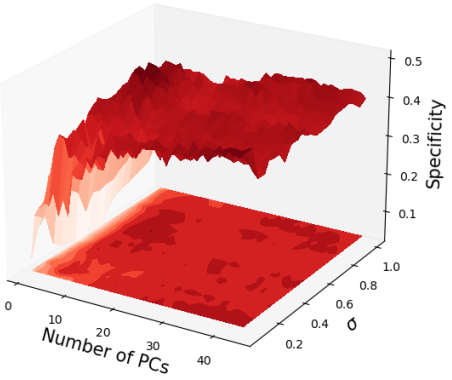
Figure 12: Performance Measures vs σ and Number of PCs using SVM (100 Trials)



(a) Accuracy vs σ and Number of PCs

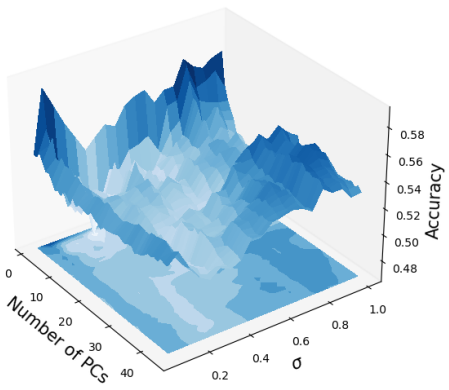


(b) Sensitivity vs σ and Number of PCs

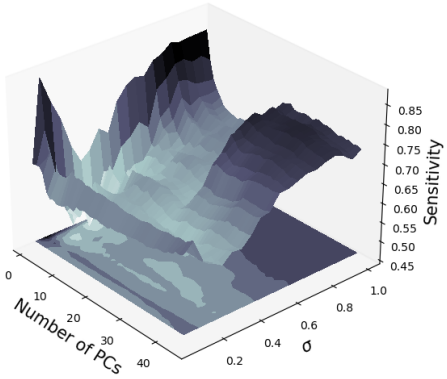


(c) Specificity vs σ and Number of PCs

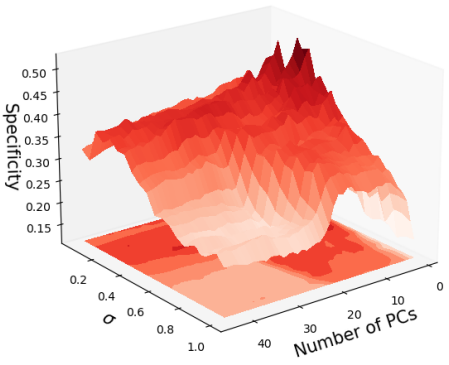
Figure 13: Performance Measures vs σ and Number of PCs using Logistic Regression (100 Trials)



(a) Accuracy vs σ and Number of PCs



(b) Sensitivity vs σ and Number of PCs

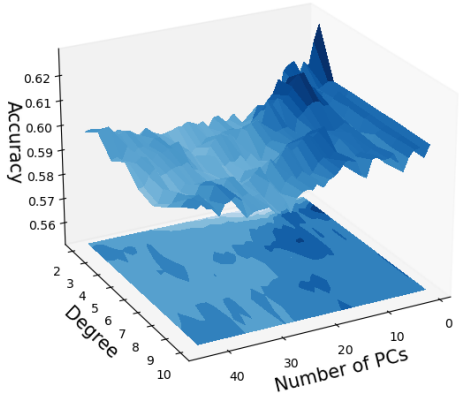


(c) Specificity vs σ and Number of PCs

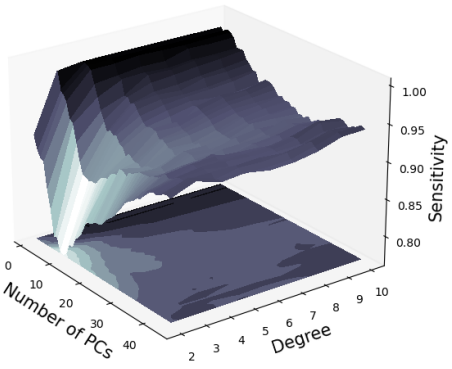
Figure 14: Performance Measures vs σ and Number of PCs using Naive Bayes (100 Trials)

5.2.2 Polynomial Kernel

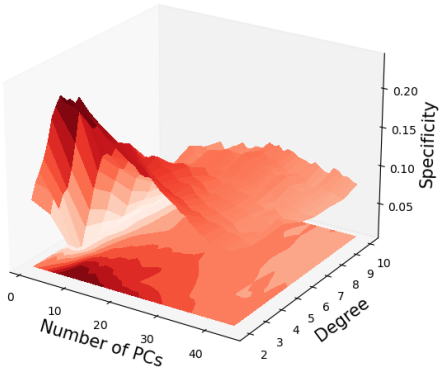
Similarly for kernel PCA with a polynomial kernel, one should investigate whether there is an ideal degree of polynomial to use as the kernel. Figures 15a - 17c illustrate this (for degrees two to ten) - where once again, no clear pattern seems to emerge with respect to the ideal number of principal components one should use. There are prominent values for the degree of the polynomials however. For accuracy and sensitivity, these degrees are $\{2, 3, 7\}$, and for specificity, they are $\{2, 5, 10\}$.



(a) Accuracy vs Degrees and Number of PCs

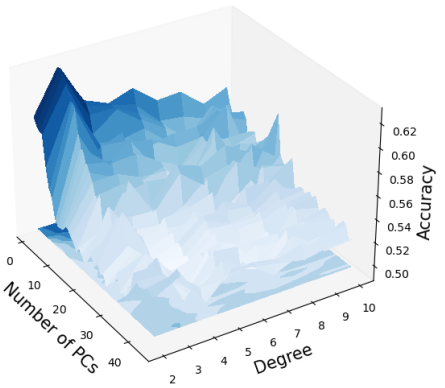


(b) Sensitivity vs Degree and Number of PCs

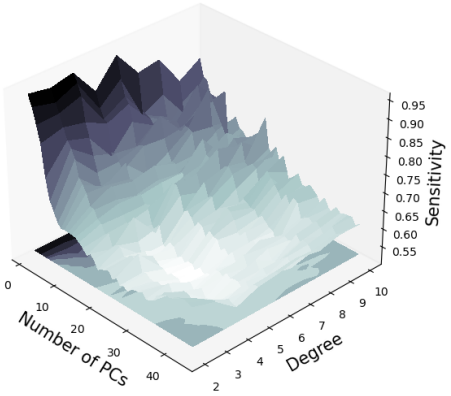


(c) Specificity vs Degree and Number of PCs

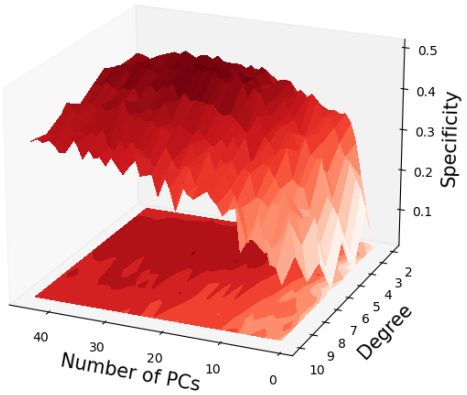
Figure 15: Performance Measures vs Degree and Number of PCs using SVM (100 Trials)



(a) Accuracy vs Degrees and Number of PCs

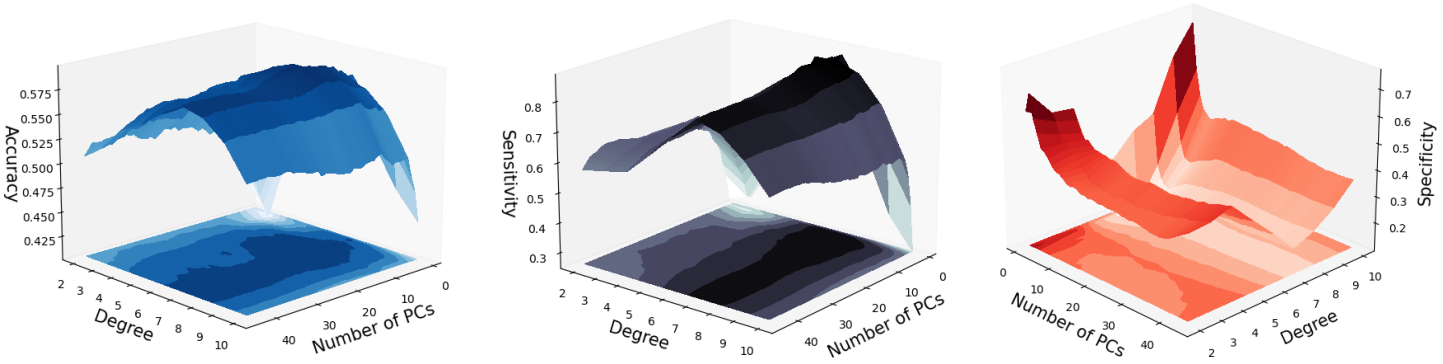


(b) Sensitivity vs Degree and Number of PCs



(c) Specificity vs Degree and Number of PCs

Figure 16: Performance Measures vs Degree and Number of PCs using Logistic Regression (100 Trials)



(a) Accuracy vs Degrees and Number of PCs

(b) Sensitivity vs Degree and Number of PCs

(c) Specificity vs Degree and Number of PCs

Figure 17: Performance Measures vs Degree and Number of PCs using Naive Bayes (100 Trials)

Furthermore, for both the use of the RBF and polynomial kernel, performance measures brought about by the noticeable values for σ and degree of polynomial are compared to linear PCA using the first eight principal components in Table 2. For the performance measure accuracy, Figure 18 shows that linear PCA performs better than kernel PCA (RBF kernel with $\sigma = 0.05$ and a quadratic polynomial kernel were deemed to give the highest accuracies, so were plotted) for all classification methods.

With respect to sensitivity, from Figure 19 and associated Table 3, one can see that one can obtain greater sensitivities using kernel PCA, mostly through the use of SVM as the classification method, as the other two methods were not as consistent in giving higher sensitivities (RBF kernel with $\sigma = 0.05$ and a 7th degree polynomial kernel were deemed to give the highest sensitivities, so were plotted).

Finally, for specificity, we can see from Figure 20 with Table 4, that linear PCA consistently gives higher results that both RBF and polynomial kernel PCA (although an instance of greater specificity does occur when a quadratic polynomial kernel is used for dimension reduction and two principal components are used). Only Naive Bayes as a classification method was plotted as this gave rise to the highest sensitivities for kernel PCA.

Now one should not be mistaken to assess kernel PCA (both with an RBF and polynomial kernel) as the more superior dimension reduction technique when compared to linear PCA in our case. Although kernel PCA does give rise to higher performance measures sometimes, and under certain conditions - one must assess overall model quality by taking all three performance measures into account. For instance, with regards to sensitivity, kernel PCA was the more superior technique (using SVM), yet under the same conditions, struggled to generate high accuracies and specificities. In fact, a high sensitivity (with low accuracy and specificity) would suggest that the model predicted most tumors to be malignant - which although would eliminate a large number of false negatives, is not a prudent choice for a model. Being such, in our study, we can conclude kernel PCA as overall inferior to linear PCA.

Now there could be multiple reasons as to why the aforementioned conclusion is true in our study. KPCA can be prone to overfitting, especially since it uses a high-dimensional feature space, KPCA may capture spurious patterns in the data. Additionally, in high-dimensional spaces, the density of the data becomes sparse, making it difficult to estimate reliable principal components. Furthermore, since KPCA is designed to capture nonlinear relationships in the data, if the data is inherently linear or contains only weak nonlinearities, KPCA may perform worse.

Table 2: Accuracy (Mean) with Varying Number of PCs and Classification Methods - for Linear and Kernel PCA for the first Eight Principal Components (100 Trials)

			σ (RBF kernel)				Degree (Polynomial kernel)		
PCs	Classification	PCA	0.05	0.1	0.15	0.5	2	3	7
1	SVM	0.763	0.608	0.609	0.609	0.601	0.588	0.629	0.610
	Logistic	0.753	0.596	0.577	0.579	0.598	0.594	0.629	0.577
	Naive	0.743	0.539	0.586	0.570	0.553	0.452	0.431	0.511
2	SVM	0.758	0.608	0.602	0.602	0.598	0.586	0.623	0.606
	Logistic	0.761	0.569	0.573	0.542	0.568	0.589	0.630	0.562
	Naive	0.744	0.539	0.559	0.526	0.520	0.427	0.411	0.577
3	SVM	0.739	0.605	0.602	0.602	0.604	0.587	0.613	0.610
	Logistic	0.752	0.564	0.560	0.540	0.555	0.586	0.626	0.560
	Naive	0.735	0.544	0.546	0.512	0.507	0.444	0.404	0.574
4	SVM	0.733	0.605	0.598	0.595	0.602	0.595	0.615	0.609
	Logistic	0.742	0.553	0.551	0.542	0.537	0.586	0.622	0.558
	Naive	0.726	0.538	0.553	0.507	0.515	0.451	0.419	0.576
5	SVM	0.727	0.604	0.598	0.600	0.599	0.582	0.607	0.604
	Logistic	0.729	0.564	0.532	0.531	0.518	0.560	0.608	0.550
	Naive	0.716	0.533	0.540	0.501	0.507	0.470	0.430	0.584
6	SVM	0.721	0.602	0.597	0.602	0.596	0.577	0.616	0.602
	Logistic	0.726	0.545	0.524	0.516	0.510	0.543	0.609	0.555
	Naive	0.714	0.534	0.539	0.505	0.508	0.482	0.448	0.579
7	SVM	0.704	0.599	0.598	0.602	0.590	0.560	0.615	0.607
	Logistic	0.710	0.559	0.531	0.525	0.508	0.537	0.607	0.559
	Naive	0.699	0.529	0.543	0.515	0.507	0.486	0.465	0.590
8	SVM	0.699	0.596	0.597	0.608	0.596	0.565	0.613	0.608
	Logistic	0.705	0.550	0.534	0.521	0.509	0.520	0.588	0.550
	Naive	0.673	0.524	0.539	0.526	0.512	0.497	0.477	0.588

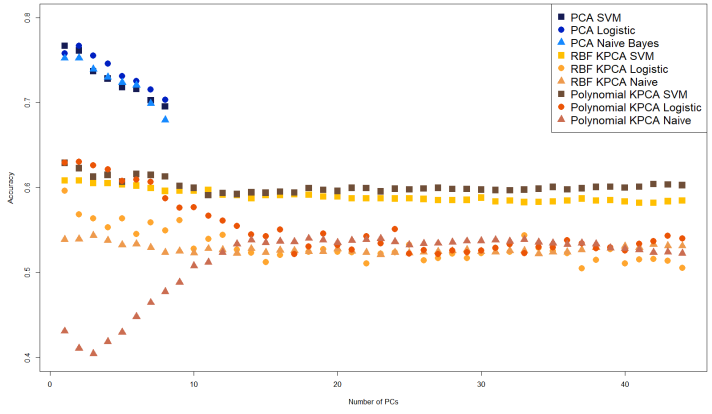


Figure 18: Accuracy (Mean) vs Number of PCs for Linear PCA, RBF KPCA ($\sigma = 0.05$) and Polynomial KPCA (degree = 3) for SVM, Logistic Regression and Naive Bayes (100 Trials)

Table 3: Sensitivity (Mean) with Varying Number of PCs and Classification Methods - for Linear and Kernel PCA for the first Eight Principal Components (100 Trials)

			σ (RBF kernel)				Degree (Polynomial kernel)		
PCs	Classification	PCA	0.05	0.1	0.15	0.5	2	3	7
1	SVM	0.884	0.981	0.988	0.989	0.965	0.909	0.986	0.990
	Logistic	0.837	0.931	0.854	0.842	0.950	0.956	0.946	0.882
	Naive	0.820	0.672	0.879	0.811	0.746	0.385	0.361	0.586
2	SVM	0.861	0.976	0.969	0.972	0.955	0.892	0.963	0.981
	Logistic	0.829	0.820	0.836	0.732	0.846	0.929	0.913	0.774
	Naive	0.819	0.674	0.761	0.636	0.635	0.284	0.310	0.872
3	SVM	0.847	0.969	0.965	0.970	0.961	0.883	0.960	0.990
	Logistic	0.819	0.799	0.795	0.713	0.774	0.887	0.890	0.778
	Naive	0.812	0.675	0.691	0.605	0.565	0.369	0.314	0.853
4	SVM	0.840	0.965	0.956	0.950	0.950	0.881	0.952	0.987
	Logistic	0.812	0.775	0.716	0.689	0.702	0.843	0.858	0.774
	Naive	0.806	0.662	0.726	0.568	0.581	0.415	0.365	0.860
5	SVM	0.829	0.960	0.949	0.960	0.942	0.847	0.931	0.975
	Logistic	0.806	0.787	0.650	0.667	0.621	0.756	0.828	0.760
	Naive	0.801	0.617	0.676	0.549	0.565	0.476	0.405	0.876
6	SVM	0.831	0.957	0.949	0.959	0.934	0.821	0.928	0.968
	Logistic	0.804	0.741	0.615	0.626	0.586	0.699	0.812	0.775
	Naive	0.806	0.606	0.655	0.564	0.579	0.497	0.431	0.859
7	SVM	0.822	0.955	0.950	0.957	0.922	0.785	0.905	0.967
	Logistic	0.771	0.786	0.632	0.636	0.589	0.675	0.788	0.794
	Naive	0.790	0.610	0.657	0.589	0.578	0.515	0.469	0.870
8	SVM	0.821	0.944	0.945	0.962	0.934	0.786	0.881	0.962
	Logistic	0.771	0.749	0.667	0.604	0.590	0.650	0.743	0.739
	Naive	0.727	0.603	0.660	0.611	0.577	0.546	0.501	0.858

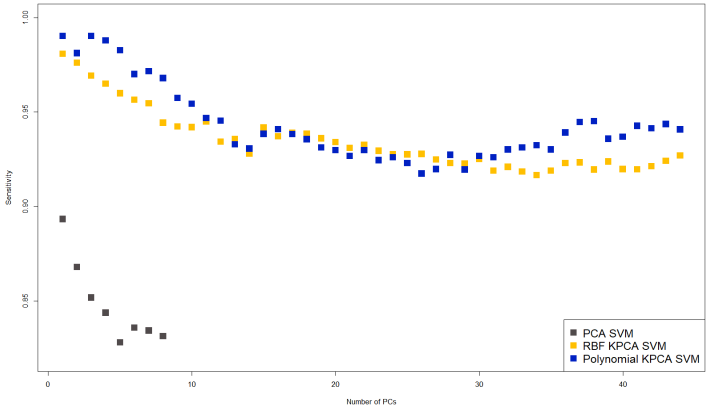


Figure 19: Sensitivity (Mean) vs Number of PCs for Linear PCA, RBF KPCA ($\sigma = 0.05$) and Polynomial KPCA (degree = 7) for SVM (100 Trials)

Table 4: Specificity (Mean) with Varying Number of PCs and Classification Methods - for Linear and Kernel PCA for the first Eight Principal Components (100 Trials)

			σ (RBF kernel)				Degree (Polynomial kernel)		
PCs	Classification	PCA	0.3	0.5	0.6	1	2	5	10
1	SVM	0.585	0.012	0.030	0.098	0.208	0.096	0.010	0.010
	Logistic	0.637	0.050	0.045	0.080	0.157	0.023	0.126	0.150
	Naive	0.643	0.373	0.257	0.204	0.157	0.613	0.366	0.733
2	SVM	0.613	0.026	0.044	0.110	0.246	0.119	0.020	0.020
	Logistic	0.672	0.195	0.150	0.156	0.250	0.055	0.170	0.188
	Naive	0.645	0.485	0.364	0.305	0.202	0.682	0.274	0.560
3	SVM	0.586	0.032	0.039	0.059	0.268	0.134	0.011	0.009
	Logistic	0.664	0.266	0.232	0.252	0.310	0.117	0.193	0.191
	Naive	0.635	0.512	0.416	0.375	0.249	0.598	0.276	0.463
4	SVM	0.580	0.037	0.059	0.080	0.267	0.152	0.013	0.011
	Logistic	0.649	0.335	0.299	0.326	0.319	0.189	0.214	0.248
	Naive	0.622	0.477	0.408	0.364	0.270	0.549	0.278	0.410
5	SVM	0.585	0.035	0.061	0.106	0.237	0.175	0.033	0.013
	Logistic	0.624	0.362	0.381	0.390	0.327	0.270	0.302	0.296
	Naive	0.604	0.459	0.418	0.378	0.280	0.499	0.276	0.395
6	SVM	0.566	0.062	0.070	0.091	0.221	0.206	0.051	0.023
	Logistic	0.620	0.403	0.414	0.426	0.325	0.315	0.298	0.244
	Naive	0.591	0.446	0.392	0.369	0.290	0.492	0.290	0.379
7	SVM	0.540	0.047	0.079	0.095	0.195	0.222	0.076	0.031
	Logistic	0.630	0.399	0.404	0.456	0.342	0.336	0.308	0.239
	Naive	0.575	0.428	0.394	0.362	0.301	0.472	0.295	0.387
8	SVM	0.528	0.049	0.067	0.096	0.189	0.235	0.093	0.039
	Logistic	0.617	0.379	0.396	0.469	0.358	0.333	0.346	0.208
	Naive	0.609	0.397	0.409	0.357	0.304	0.449	0.297	0.396

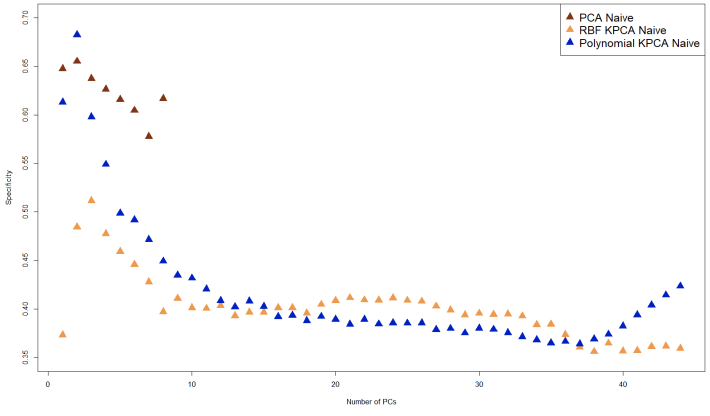


Figure 20: Specificity (Mean) vs Number of PCs for Linear PCA, RBF KPCA ($\sigma = 0.3$) and Polynomial KPCA (degree = 2) for Naive Bayes (100 Trials)

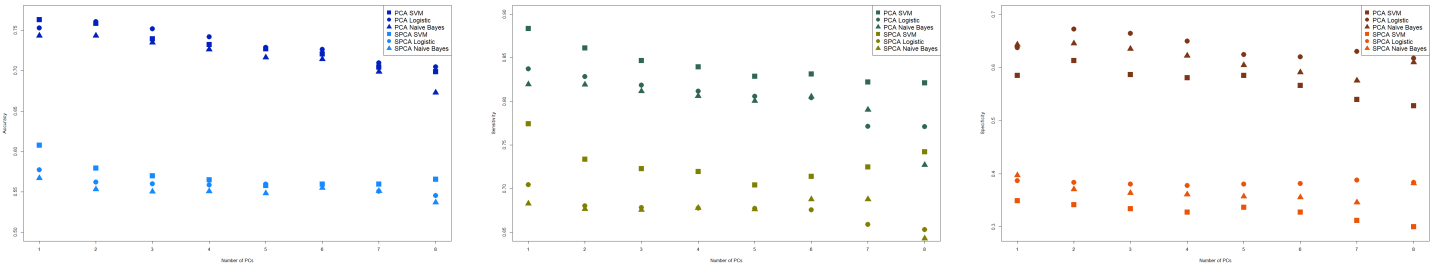
5.3 Sparse Principal Component Analysis

In similar fashion, we aim to determine if there are a number of sparse principal components to be used in dimension reduction, which would bring about the highest performance measures. But moreover, if utilising SPCA is worthwhile in comparison to linear PCA - established now as the benchmark dimension reduction technique. From Section 4.1.3 we know Sparse PCA tends to be more robust to noise and redundant features compared to PCA. By encouraging sparsity in the loadings of the principal components, Sparse PCA can filter out noisy or irrelevant features, leading to a more robust representation of the underlying data structure. From Figures 21a - 21c with corresponding Table 5, we clearly see SPCA's inferior performance relative to PCA with respect to all performance measures (and across all classification methods). We notice the similar nature to PCA, where there is a decreased performance with increasing the number of PCs used. One could argue that different penalty term control parameters, α and β (as discussed in Section 4.1.3) could result in different results - this study does not investigate further with such parameter tuning.

Now, SPCA performing worse than PCA in our study may be because of a loss of information. We know Sparse PCA encourages sparsity in the loadings of the principal components, which means that only a subset of the original features contribute significantly to each principal component. This sparsity may lead to a loss of important features which are excluded from the sparse representation. In contrast, PCA considers all features equally in each principal component and may provide a more balanced representation of the data.

Table 5: Performance Measures (Mean) with Varying Number of PCs and Classification Methods for PCA and SPCA (1000 Trials)

		Accuracy		Sensitivity		Specificity	
PCs	Classification	PCA	SPCA	PCA	SPCA	PCA	SPCA
1	SVM	0.763	0.608	0.884	0.774	0.585	0.349
	Logistic	0.753	0.577	0.837	0.704	0.637	0.387
	Naive	0.743	0.567	0.820	0.683	0.643	0.397
2	SVM	0.758	0.579	0.861	0.734	0.613	0.342
	Logistic	0.761	0.562	0.829	0.680	0.672	0.384
	Naive	0.744	0.553	0.819	0.676	0.645	0.371
3	SVM	0.739	0.570	0.847	0.723	0.586	0.334
	Logistic	0.752	0.560	0.819	0.678	0.664	0.380
	Naive	0.735	0.550	0.812	0.676	0.635	0.364
4	SVM	0.733	0.565	0.840	0.720	0.580	0.328
	Logistic	0.742	0.558	0.812	0.678	0.649	0.378
	Naive	0.726	0.551	0.806	0.678	0.622	0.361
5	SVM	0.727	0.558	0.829	0.704	0.585	0.337
	Logistic	0.729	0.560	0.806	0.677	0.624	0.380
	Naive	0.716	0.548	0.801	0.676	0.604	0.357
6	SVM	0.721	0.560	0.831	0.714	0.566	0.328
	Logistic	0.726	0.559	0.804	0.676	0.620	0.382
	Naive	0.714	0.555	0.806	0.688	0.591	0.356
7	SVM	0.704	0.560	0.822	0.725	0.540	0.312
	Logistic	0.710	0.551	0.771	0.659	0.630	0.388
	Naive	0.699	0.551	0.790	0.688	0.575	0.346
8	SVM	0.699	0.566	0.821	0.742	0.528	0.300
	Logistic	0.705	0.545	0.771	0.653	0.617	0.384
	Naive	0.673	0.537	0.727	0.643	0.609	0.382



(a) Accuracy vs Number of PCs (b) Sensitivity vs Number of PCs (c) Specificity vs Number of PCs

Figure 21: Performance Measures vs Number of PCs for PCA and SPCA for all Classification Methods (1000 Trials)

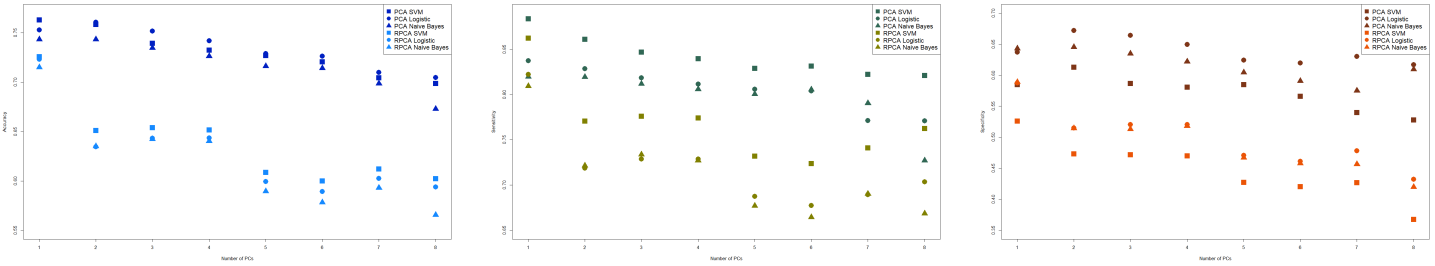
5.4 Robust Principal Component Analysis

With the aim of combating against outliers (potential "corrupted" observations or gross errors in the data) having a significant effect on our model building, we employ robust PCA to counter potential outliers, as expounded upon in Section 4.1.4. As before, we aim to determine if there are a number of robust principal components to be used in dimension reduction, which would bring about the highest performance measures, but more importantly, if RPCA gives rise to better performance measures relative to PCA. From Figures 22a - 22c with corresponding Table 6, we see RPCA's inferior performance relative to PCA with respect to all performance measures (and across all classification methods) for the default robustness parameter $\alpha = 0.75$. We notice in Table 6, that with an $\alpha = 1$ (maximum robustness by downweighting the influence of outliers as much as possible), utilising the first principal component gives similar performance measures relative to linear PCA - performance measures fall off greatly after this however. Additionally, for default robustness parameter, we notice the similar nature to PCA, where there is a decreased performance with increasing the number of PCs.

We argue that RPCA's general inferiority relative to PCA (apart from utilising the first PC with $\alpha = 1$) is due to a loss of information from the data - robust estimators may downweight or exclude certain observations that are considered outliers, leading to a loss of information or distortion of the underlying structure. While this loss of information may improve robustness to outliers, it could also result in reduced fidelity or representativeness of the estimated principal components compared to PCA.

Table 6: Performance Measures (Mean) with Varying Number of PCs and Classification Methods for PCA and RPCA for Default Robustness Parameter $\alpha_{0.75} = 0.75$ and $\alpha_1 = 1$ (1000 Trials)

		Accuracy			Sensitivity			Specificity		
PCs		PCA	RPCA$^{(\alpha_{0.75})}$	RPCA$^{(\alpha_1)}$	PCA	RPCA$^{(\alpha_{0.75})}$	RPCA$^{(\alpha_1)}$	PCA	RPCA$^{(\alpha_{0.75})}$	RPCA$^{(\alpha_1)}$
1	SVM	0.767	0.720	0.771	0.893	0.868	0.895	0.588	0.511	0.601
	Logistic	0.758	0.719	0.767	0.852	0.825	0.855	0.635	0.580	0.656
	Naive	0.752	0.706	0.756	0.836	0.808	0.838	0.647	0.575	0.659
2	SVM	0.761	0.679	0.642	0.868	0.800	0.767	0.617	0.510	0.464
	Logistic	0.767	0.660	0.627	0.835	0.748	0.719	0.686	0.543	0.503
	Naive	0.752	0.658	0.629	0.833	0.752	0.722	0.655	0.535	0.504
3	SVM	0.737	0.669	0.674	0.852	0.798	0.802	0.583	0.487	0.494
	Logistic	0.755	0.653	0.669	0.828	0.743	0.758	0.666	0.531	0.546
	Naive	0.740	0.655	0.659	0.823	0.755	0.756	0.637	0.522	0.531
4	SVM	0.728	0.682	0.701	0.844	0.809	0.821	0.574	0.503	0.536
	Logistic	0.746	0.676	0.694	0.817	0.761	0.777	0.659	0.561	0.586
	Naive	0.730	0.674	0.690	0.815	0.763	0.777	0.626	0.558	0.577
5	SVM	0.718	0.636	0.692	0.828	0.760	0.806	0.575	0.462	0.538
	Logistic	0.731	0.622	0.696	0.809	0.713	0.776	0.631	0.496	0.591
	Naive	0.724	0.618	0.686	0.812	0.707	0.763	0.616	0.500	0.590
6	SVM	0.716	0.597	0.692	0.836	0.729	0.816	0.558	0.413	0.527
	Logistic	0.725	0.604	0.693	0.804	0.695	0.779	0.625	0.475	0.578
	Naive	0.720	0.582	0.677	0.813	0.671	0.770	0.604	0.461	0.554
7	SVM	0.703	0.602	0.705	0.834	0.738	0.837	0.530	0.412	0.528
	Logistic	0.715	0.582	0.699	0.776	0.672	0.777	0.646	0.458	0.594
	Naive	0.699	0.589	0.693	0.795	0.686	0.802	0.578	0.454	0.545
8	SVM	0.696	0.610	0.677	0.831	0.782	0.812	0.517	0.370	0.494
	Logistic	0.703	0.605	0.674	0.776	0.721	0.758	0.615	0.441	0.560
	Naive	0.679	0.579	0.639	0.738	0.687	0.700	0.617	0.433	0.570



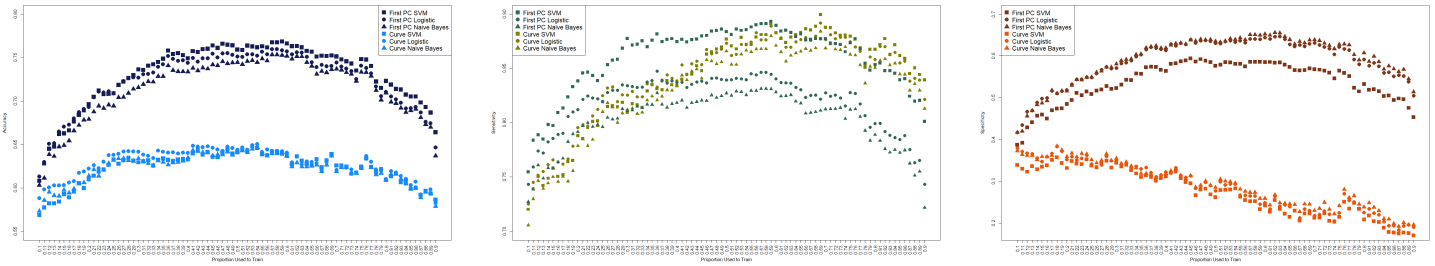
(a) Accuracy vs Number of PCs (b) Sensitivity vs Number of PCs (c) Specificity vs Number of PCs

Figure 22: Performance Measures vs Number of PCs for PCA and RPCA for all Classification Methods for Default Robustness $\alpha = 0.75$ (1000 Trials)

5.5 Principal Curves

At the conclusion of Section 5.1, we deduced that an increase in the number of principal components used as a dimension reduction technique (in both training and testing phase) gave rise to lower performance measures. This infers that solely using a single (the first) principal component would suffice in training the prognosis model.

Now seeing as principal curves are somewhat analogous to PCA as explained in Section 4.1.5 i.e. the resulting feature is the 'projection' of the observation onto the principal curve, we find it fitting to compare this to utilising the first principal component obtained when conducting PCA. Furthermore, Figures 23a - 23c illustrate the different performance measures for both using a principal curve and the first principal component. We note that in general, using the first principal component is superior to using a principal curve. With regard to sensitivity however, there does seem to be times where the principal curve outperforms using the first principal component - though choosing to use the former over the latter would be futile, seeing as this is not the case with regard to accuracy and specificity.



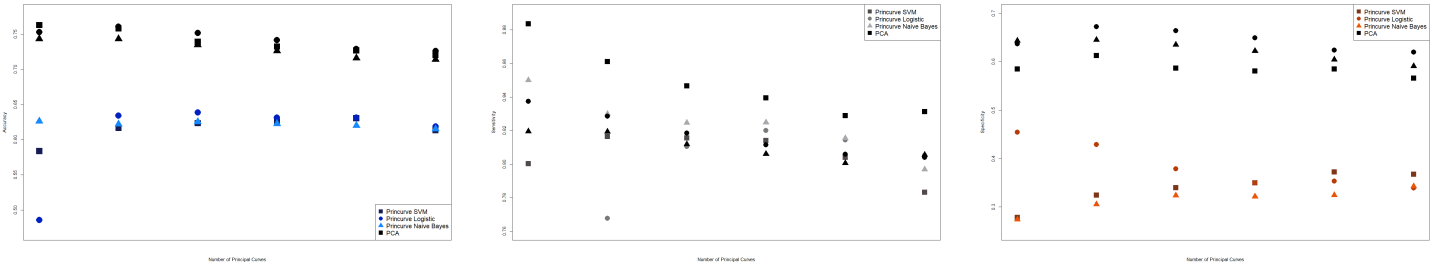
(a) Accuracy vs Train-Test Split (b) Sensitivity vs Train-Test Split (c) Specificity vs Train-Test Split

Figure 23: Performance Measures vs Train-Test Split for Principal Curve and First PC for all Classification Methods (250 Trials)

5.5.1 K-Means + Principal Curves

The authors propose a hybrid approach by utilising K-Means clustering coupled with a principal curve analysis in an attempt to enhance model predictive performance. We increase the number of principle curves on the given feature space, by initially partitioning the observations into clusters, after which, a principal curve is fit to each set of observations belonging to a cluster, where the resulting feature is the principle curve projections belonging to each set of clustered observations.

From Figures 24a - 24c, we only notice that this approach improves the accuracy, where this seems not to be the case for sensitivity and specificity. Regardless, said approach still results in inferior predictive performance of the prognosis model relative to PCA, also seen in the figures.

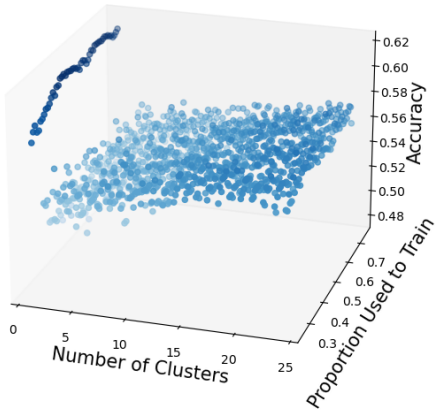


(a) Accuracy vs Number of Principal Curves (b) Sensitivity vs Number of Principal Curves (c) Specificity vs Number of Principal Curves

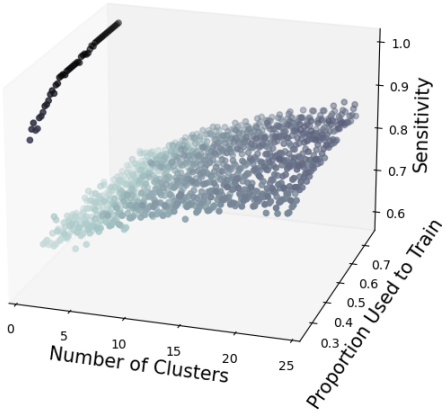
Figure 24: Performance Measures vs Number of Principal Curves all Classification Methods (100 Trials)

5.6 Cluster Analysis

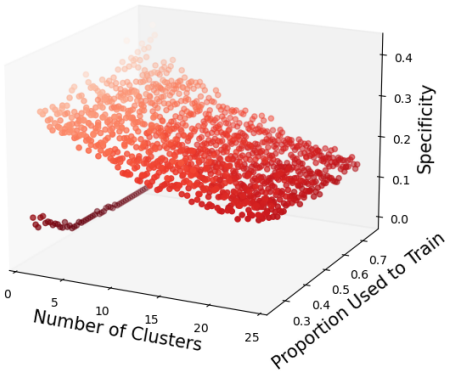
As expounded on in Section 4.5.2, we utilise cluster analysis as a method of dimension reduction - where the new feature is the cluster to which the specific observation belongs, similarly to what was undergone in [12] on the WBC dataset. With cluster analysis we aim to determine whether there are an ideal number of clusters to be used to train the model in the training phase, as well as dimension reduction used on the training set in the testing phase. Furthermore, we aim to investigate whether there is a preferred clustering method (within hierarchical and partitioning clustering). Initially, as before, we deem determining if there is an ideal train-test split, to be a fruitful task. Subsequently, Figures 25a - 27c illustrate the behaviour of performance measures against number of k-means clusters used and train-test split for all classification methods.



(a) Accuracy vs Number of Clusters and Train-Test Split

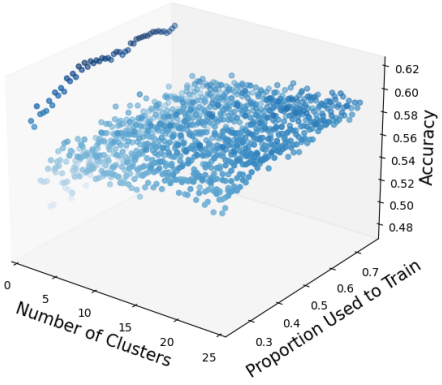


(b) Sensitivity vs Number of Clusters and Train-Test Split

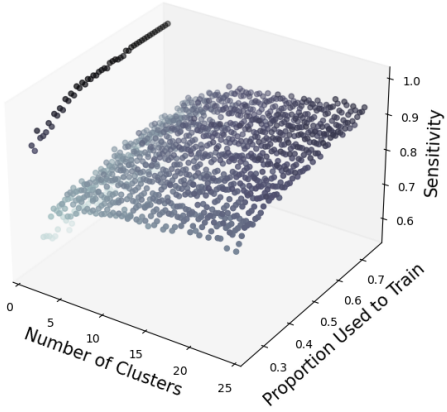


(c) Specificity vs Number of Clusters and Train-Test Split

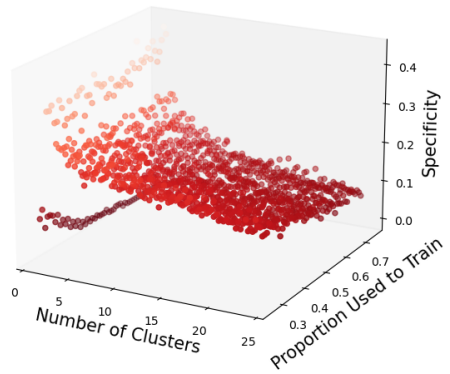
Figure 25: Performance Measures vs Number of Clusters and Train-Test Split using SVM and K-Means Clustering (200 Trials)



(a) Accuracy vs Number of Clusters and Train-Test Split

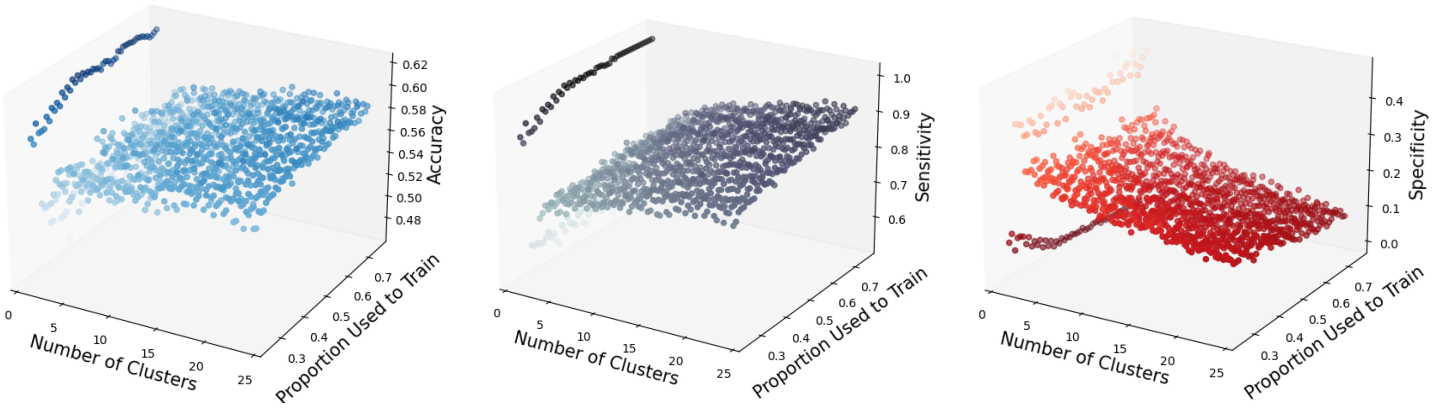


(b) Sensitivity vs Number of Clusters and Train-Test Split



(c) Specificity vs Number of Clusters and Train-Test Split

Figure 26: Performance Measures vs Number of Clusters and Train-Test Split using Logistic Regression and K-Means Clustering (200 Trials)



(a) Accuracy vs Number of Clusters and Train-Test Split

(b) Sensitivity vs Number of Clusters and Train-Test Split

(c) Specificity vs Number of Clusters and Train-Test Split

Figure 27: Performance Measures vs Number of Clusters and Train-Test Split using Naive Bayes and K-Means Clustering (200 Trials)

Now seeing as when using a single cluster, the model classifies all tumors as malignant (since there is only one grouping, and malignant tumors are the dominant class), we note the sensitivity of one in all the sensitivity plots (Figures 25b, 26b, 27b) - seeing as there are only false positives and true positives. Conversely, we observe the specificity value of zero (when using a single cluster) - this being due to there being no true negatives (the model did not predict any tumors to be benign). We notice the similar shapes for the three different classification methods, where for accuracy and sensitivity - there is an increase in the performance measure as the number of clusters increase (starting from using two clusters) where the minimum value obtained is when using two clusters. The converse is true for specificity. We note there not to be a difference in performance measure as we vary the train-test split - hence we fix this to 55/45 (as done previously in our study) and investigate further for both hierarchical and partitioning clustering techniques.

From Figures 28a - 30c (with LOESS smoothing), with corresponding Tables 7 - 9 we note a similar behaviour to what has been previously discussed with regard to performance measures when solely using a single cluster i.e. sensitivities and specificities being approximately one and zero respectively. We note that said values are somewhat all alike for all the clustering methods used. Unlike before however, different minimums and maximums of performance measures are achieved and are unique for each clustering method (with K-means: the minimum values for accuracy and sensitivity, and the maximum value for specificity was achieved when using two clusters) - where accuracy and sensitivity decrease to this minimum and then increase thereafter (the opposite for specificity). From the plots, we can infer this min/max value to be when the number of clusters in use is between two and ten (specific for each clustering method).

Table 7: Accuracy (Mean) using Ten Clusters for Agnes (Single, Average and Complete Linkage), Diana, K-Means, Pam (No Swapping and Swapping), Fanny and Model-Based Clustering Techniques using SVM, Logistic Regression and Naive Bayes (100 Trials)

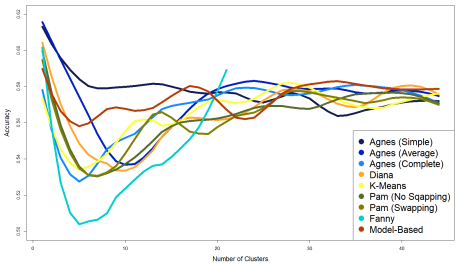
Clusters	Classification	Agnes^{S}	Agnes^{A}	Agnes^{C}	Diana	K-Means	Pam^{NS}	Pam^{S}	Fanny	Model-Based
1	SVM	0.610	0.610	0.610	0.610	0.610	0.610	0.610	0.621	0.610
	Logistic	0.610	0.610	0.610	0.610	0.610	0.610	0.610	0.622	0.610
	Naive	0.610	0.610	0.610	0.610	0.610	0.610	0.610	0.622	0.610
2	SVM	0.608	0.610	0.526	0.588	0.505	0.564	0.562	0.535	0.558
	Logistic	0.608	0.596	0.527	0.586	0.495	0.564	0.562	0.518	0.553
	Naive	0.400	0.438	0.400	0.493	0.485	0.553	0.551	0.514	0.545
3	SVM	0.599	0.600	0.524	0.560	0.546	0.563	0.543	0.509	0.554
	Logistic	0.602	0.599	0.522	0.562	0.551	0.573	0.561	0.511	0.586
	Naive	0.471	0.505	0.471	0.546	0.560	0.565	0.552	0.511	0.579
4	SVM	0.590	0.584	0.525	0.542	0.533	0.532	0.532	0.511	0.554
	Logistic	0.610	0.592	0.544	0.557	0.559	0.577	0.572	0.508	0.564
	Naive	0.534	0.562	0.534	0.550	0.549	0.558	0.554	0.507	0.564
5	SVM	0.571	0.577	0.527	0.554	0.546	0.540	0.544	0.511	0.564
	Logistic	0.607	0.584	0.567	0.569	0.577	0.572	0.581	0.496	0.583
	Naive	0.559	0.575	0.559	0.557	0.568	0.560	0.566	0.491	0.581
6	SVM	0.591	0.560	0.522	0.566	0.540	0.530	0.542	0.500	0.550
	Logistic	0.591	0.581	0.579	0.580	0.574	0.578	0.586	0.517	0.572
	Naive	0.572	0.575	0.572	0.578	0.552	0.561	0.578	0.522	0.572
7	SVM	0.577	0.557	0.551	0.521	0.537	0.523	0.530	0.516	0.571
	Logistic	0.592	0.572	0.594	0.573	0.564	0.564	0.568	0.526	0.575
	Naive	0.569	0.568	0.569	0.551	0.548	0.544	0.546	0.530	0.573
8	SVM	0.577	0.544	0.548	0.532	0.541	0.543	0.529	0.500	0.573
	Logistic	0.595	0.584	0.579	0.580	0.557	0.558	0.573	0.511	0.576
	Naive	0.582	0.563	0.582	0.558	0.548	0.536	0.552	0.521	0.568
9	SVM	0.577	0.546	0.556	0.546	0.538	0.534	0.533	0.515	0.567
	Logistic	0.602	0.584	0.584	0.590	0.586	0.563	0.569	0.492	0.580
	Naive	0.581	0.553	0.581	0.562	0.569	0.542	0.555	0.491	0.576
10	SVM	0.584	0.522	0.542	0.533	0.565	0.537	0.537	0.531	0.568
	Logistic	0.600	0.591	0.587	0.585	0.571	0.567	0.576	0.518	0.587
	Naive	0.581	0.551	0.581	0.558	0.573	0.544	0.554	0.517	0.581

Table 8: Sensitivity (Mean) using Ten Clusters for Agnes (Single, Average and Complete Linkage), Diana, K-Means, Pam (No Swapping and Swapping), Fanny and Model-Based Clustering Techniques using SVM, Logistic Regression and Naive Bayes (100 Trials)

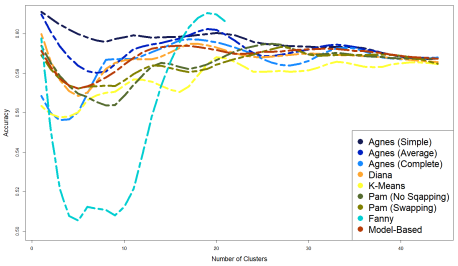
Clusters	Classification	$Agnes^S$	$Agnes^A$	$Agnes^C$	Diana	K-Means	Pam^{NS}	Pam^S	Fanny	Model-Based
1	SVM	0.990	0.990	0.990	0.990	0.990	0.990	0.990	1	0.990
	Logistic	0.990	0.990	0.990	0.990	0.990	0.990	0.990	1	0.990
	Naive	0.990	0.990	0.990	0.990	0.990	0.990	0.990	1	0.990
2	SVM	0.979	0.990	0.707	0.915	0.668	0.813	0.768	0.584	0.815
	Logistic	0.979	0.942	0.697	0.900	0.636	0.813	0.768	0.612	0.797
	Naive	0.051	0.205	0.051	0.570	0.577	0.778	0.746	0.573	0.745
3	SVM	0.942	0.955	0.745	0.835	0.683	0.722	0.695	0.617	0.746
	Logistic	0.951	0.949	0.712	0.813	0.773	0.761	0.758	0.614	0.844
	Naive	0.440	0.581	0.440	0.753	0.767	0.752	0.753	0.599	0.815
4	SVM	0.890	0.916	0.736	0.762	0.703	0.692	0.696	0.576	0.749
	Logistic	0.977	0.929	0.775	0.754	0.802	0.823	0.811	0.634	0.808
	Naive	0.637	0.770	0.637	0.761	0.775	0.772	0.765	0.610	0.791
5	SVM	0.824	0.845	0.710	0.767	0.707	0.698	0.695	0.609	0.757
	Logistic	0.957	0.883	0.813	0.787	0.814	0.821	0.828	0.582	0.848
	Naive	0.780	0.789	0.780	0.771	0.787	0.781	0.800	0.566	0.817
6	SVM	0.922	0.792	0.707	0.735	0.696	0.684	0.720	0.619	0.776
	Logistic	0.915	0.866	0.840	0.803	0.832	0.837	0.851	0.586	0.850
	Naive	0.831	0.810	0.831	0.780	0.789	0.806	0.830	0.585	0.851
7	SVM	0.877	0.793	0.771	0.688	0.729	0.707	0.728	0.595	0.795
	Logistic	0.918	0.841	0.884	0.832	0.831	0.839	0.856	0.628	0.861
	Naive	0.801	0.788	0.801	0.802	0.797	0.811	0.818	0.623	0.851
8	SVM	0.849	0.736	0.764	0.705	0.739	0.759	0.732	0.583	0.806
	Logistic	0.923	0.854	0.863	0.865	0.840	0.854	0.868	0.634	0.862
	Naive	0.851	0.797	0.851	0.807	0.816	0.810	0.835	0.630	0.836
9	SVM	0.843	0.746	0.773	0.727	0.744	0.764	0.752	0.610	0.811
	Logistic	0.939	0.867	0.879	0.882	0.862	0.859	0.863	0.546	0.857
	Naive	0.829	0.787	0.829	0.816	0.829	0.823	0.848	0.536	0.834
10	SVM	0.871	0.704	0.772	0.734	0.771	0.770	0.758	0.631	0.814
	Logistic	0.926	0.886	0.880	0.887	0.848	0.869	0.868	0.643	0.875
	Naive	0.844	0.805	0.844	0.822	0.837	0.829	0.843	0.638	0.849

Table 9: Specificity (Mean) using Ten Clusters for Agnes (Single, Average and Complete Linkage), Diana, K-Means, Pam (No Swapping and Swapping), Fanny and Model-Based Clustering Techniques using SVM, Logistic Regression and Naive Bayes (100 Trials)

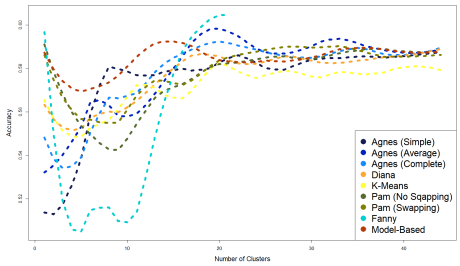
Clusters	Classification	$Agnes^S$	$Agnes^A$	$Agnes^C$	Diana	K-Means	Pam^{NS}	Pam^S	Fanny	Model-Based
1	SVM	0.010	0.010	0.010	0.010	0.010	0.010	0.010	0	0.010
	Logistic	0.010	0.010	0.010	0.010	0.010	0.010	0.010	0	0.010
	Naive	0.010	0.010	0.010	0.010	0.010	0.010	0.010	0	0.010
2	SVM	0.020	0.010	0.276	0.084	0.294	0.203	0.266	0.476	0.182
	Logistic	0.020	0.059	0.293	0.102	0.312	0.203	0.266	0.392	0.197
	Naive	0.950	0.800	0.950	0.394	0.368	0.226	0.273	0.435	0.256
3	SVM	0.061	0.040	0.198	0.153	0.328	0.323	0.308	0.364	0.281
	Logistic	0.050	0.050	0.247	0.191	0.211	0.291	0.270	0.377	0.197
	Naive	0.548	0.394	0.548	0.234	0.242	0.286	0.256	0.394	0.229
4	SVM	0.111	0.078	0.213	0.214	0.278	0.301	0.287	0.432	0.268
	Logistic	0.020	0.071	0.195	0.262	0.203	0.207	0.210	0.329	0.208
	Naive	0.355	0.228	0.355	0.234	0.222	0.247	0.245	0.364	0.229
5	SVM	0.182	0.160	0.245	0.231	0.307	0.300	0.320	0.375	0.275
	Logistic	0.039	0.125	0.190	0.241	0.224	0.199	0.217	0.380	0.189
	Naive	0.213	0.237	0.213	0.236	0.248	0.242	0.224	0.388	0.234
6	SVM	0.077	0.196	0.247	0.325	0.322	0.300	0.281	0.344	0.217
	Logistic	0.084	0.145	0.177	0.236	0.191	0.193	0.195	0.416	0.160
	Naive	0.176	0.204	0.176	0.274	0.207	0.200	0.213	0.428	0.151
7	SVM	0.127	0.196	0.208	0.289	0.265	0.255	0.241	0.408	0.243
	Logistic	0.085	0.172	0.138	0.194	0.161	0.150	0.130	0.395	0.147
	Naive	0.204	0.231	0.204	0.186	0.179	0.151	0.149	0.409	0.158
8	SVM	0.154	0.246	0.222	0.279	0.258	0.217	0.232	0.384	0.222
	Logistic	0.078	0.171	0.141	0.156	0.134	0.112	0.122	0.335	0.142
	Naive	0.157	0.214	0.157	0.190	0.152	0.131	0.131	0.362	0.166
9	SVM	0.167	0.241	0.226	0.282	0.247	0.190	0.207	0.379	0.210
	Logistic	0.069	0.153	0.126	0.141	0.170	0.122	0.125	0.423	0.161
	Naive	0.193	0.210	0.193	0.189	0.184	0.128	0.121	0.434	0.187
10	SVM	0.137	0.253	0.195	0.248	0.271	0.188	0.207	0.402	0.213
	Logistic	0.084	0.140	0.139	0.122	0.158	0.114	0.137	0.351	0.146
	Naive	0.177	0.177	0.177	0.166	0.182	0.124	0.128	0.352	0.173



(a) Accuracy vs Number of Clusters using SVM

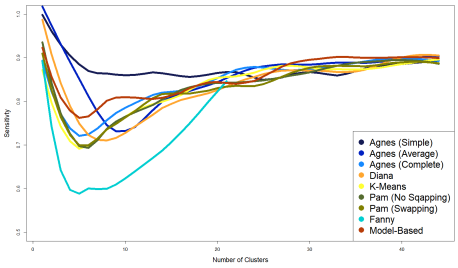


(b) Accuracy vs Number of Clusters using Logistic Regression

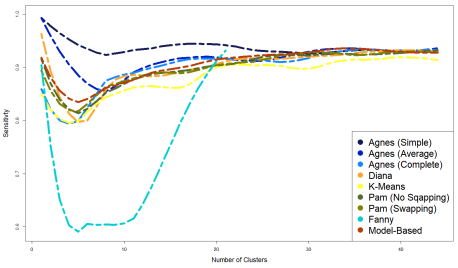


(c) Accuracy vs Number of Clusters using Naive Bayes

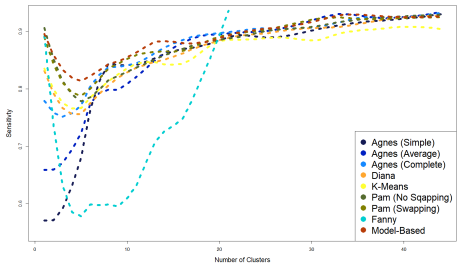
Figure 28: Accuracy vs Number of Clusters for Agnes (Single, Average and Complete Linkage), Diana, K-Means, Pam (No Swapping and Swapping), Fanny and Model-Based Clustering Techniques for all Classification Methods (100 Trials) with LOESS Smoothing



(a) Sensitivity vs Number of Clusters using SVM

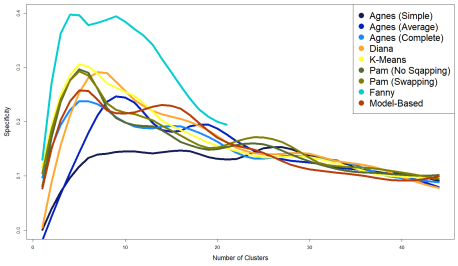


(b) Sensitivity vs Number of Clusters using Logistic Regression

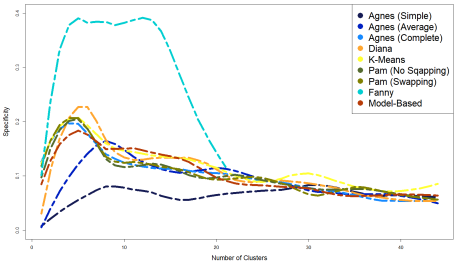


(c) Sensitivity vs Number of Clusters using Naive Bayes

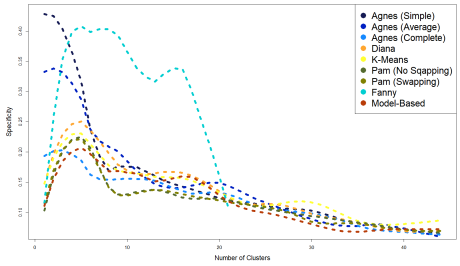
Figure 29: Sensitivity vs Number of Clusters for Agnes (Single, Average and Complete Linkage), Diana, K-Means, Pam (No Swapping and Swapping), Fanny and Model-Based Clustering Techniques for all Classification Methods (100 Trials) with LOESS Smoothing



(a) Specificity vs Number of Clusters using SVM



(b) Specificity vs Number of Clusters using Logistic Regression



(c) Specificity vs Number of Clusters using Naive Bayes

Figure 30: Specificity vs Number of Clusters for Agnes (Single, Average and Complete Linkage), Diana, K-Means, Pam (No Swapping and Swapping), Fanny and Model-Based Clustering Techniques for all Classification Methods (100 Trials) with LOESS Smoothing

Furthermore, we notice that the partitioning clustering method, fanny, seems to give rise to performance measures which are more extreme relative to other clustering techniques (lower accuracy and sensitivity, and higher specificity). Additionally, there seems not to be an obvious choice of clustering technique nor classification method to utilize - that is, no single clustering method nor classification method performs better than the rest. Moreover, the ideal choice for the number of clusters to use seems illusive - if one chooses to maximise accuracy and sensitivity, this would be at the expense of specificity (and vice versa).

Now with regard to the comparison to PCA - we can infer that clustering as a form of dimension reduction does not 'hold a candle' to PCA, especially with regard to accuracy and specificity - contrary to [12]'s

work on the WBC dataset. No clustering technique gave rise to a higher accuracy nor specificity when compared to PCA. Now although sensitivity values were comparable, a high sensitivity coupled with low accuracy and specificity would imply, as expounded on before, that the model characterized most tumors to be malignant - an imprudent approach to modeling.

6 Conclusion

The study hypothesised that there would be a superior dimension reduction technique for training a model to predict a prostate tumor to be malignant or benign. We further posited that said dimension reduction would perform adequately against a model built on the entire feature set.

Subsequently, the study embarked on a search for this dimension reduction method through the in-depth analysis of linear, kernel, sparse and robust principal component analysis, including principal curves and cluster analysis (an unconventional means of dimension reduction). This resulted in us concluding that the popular, yet mundane, linear principal component analysis was all that was necessary to fill the bill. Additionally, the study determined that one would need only to use the first principal component to train a tumor classification model, as this would yield the best predictive performance (based on accuracy, sensitivity and specificity).

Now training a model on the full feature set (a total of eight features) resulted in an accuracy of approximately 81%, a sensitivity of 89%, and a specificity of 70% (using SVM-RBF kernel). Comparatively, PCA yielded an accuracy of approximately 76%, a sensitivity of 88%, and a specificity of 59% (also SVM-RBF kernel) just from solely using the first principal component to build the model. We view this reduction of the eight features down to a single feature all the while obtaining these exceptional performance metrics to be quite exemplary in the authors' opinion.

Additionally, the notion of one of the classification methods used in the study namely SVM-RBF kernel, Logistic Regression and Naive Bayes, being superior to the others, remains unclear. We posit that future studies allow for parameter tuning of their classification methods: tuning the cost and γ parameters in the R package svm (with RBF kernel), or employing elastic net or LASSO logistic regression. Other classification algorithms could be utilised, popular ones include Random Forest, Decision tree and K-nearest Neighbours as used in [1] and [4].

With regards to additional dimension reduction techniques, utilising manifold learning techniques or auto encoders may be a fruitful task, though extensive tuning of hyper parameters may have to be undergone. If our study had eluded to there being nonlinear relationships in our data, utilising the aforementioned dimension reduction methods would be worthwile.

Furthermore, our study utilised repeated corss-validation to obtain more reliable estimates of the performance metric, yet with the presence of a class imbalance (malignant tumors constituted 62% of the observations), stratified cross-validation may have been a more prudent approach to retain said class imbalance. Future studies should also maybe assess other performance metrics as expounded on in Section 4.3.

References

[1] Nyme Ahmed, Syed Nafiul Shefat, et al. Performance evaluation of data mining classification algorithms for predicting breast cancer. *Malaysian Journal of Science and Advanced Technology*, pages 90–95, 2022.

[2] Andriy Burkov. *The hundred-page machine learning book*, volume 1. Andriy Burkov Quebec City, QC, Canada, 2019.

[3] Marie Chavent, Vanessa Kuentz, Benoˆıt Liquet, and L Saracco. Clustofvar: An r package for the clustering of variables. *arXiv preprint arXiv:1112.0295*, 2011.

[4] Chitra Desai. Analysis of impact of principal component analysis and feature selection for detection of breast cancer using machine learning algorithms. *Journal Name*, 13:197–221, 01 2023.

[5] Men's Foundation. Prostate cancer, n.d. Accessed: March 19, 2024.

[6] Surbhi Gupta, Manoj Kumar Gupta, and Rakesh Kumar. A novel multi-neural ensemble approach for cancer diagnosis. *Applied Artificial Intelligence*, 36(1):2018182, 2022.

[7] Annisa Handayani, Ade Jamal, and Ali Akbar Septiandri. Evaluasi tiga jenis algoritme berbasis pembelajaran mesin untuk klasifikasi jenis tumor payudara. *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, 6(4):394–403, 2017.

[8] Trevor Hastie and Werner Stuetzle. Principal curves. *Journal of the American Statistical Association*, 84(406):502–516, 1989.

[9] Heiko Hoffmann. Kernel pca for novelty detection. *Pattern recognition*, 40(3):863–874, 2007.

[10] Mia Hubert, Peter J Rousseeuw, and Karlien Vanden Branden. Robpca: a new approach to robust principal component analysis. *Technometrics*, 47(1):64–79, 2005.

[11] Sara Ibrahim, Saima Nazir, and Sergio A Velastin. Feature selection using correlation analysis and principal component analysis for accurate breast cancer diagnosis. *Journal of imaging*, 7(11):225, 2021.

[12] Ade Jamal, Annisa Handayani, Ali Akbar Septiandri, Endang Ripmiatin, and Yunus Effendi. Dimensionality reduction using pca and k-means clustering for breast cancer prediction. *Lontar Komput. J. Ilm. Teknol. Inf*, 9(3):192–201, 2018.

[13] Shubhangi N Katole and Swapnili P Karmore. A new approach of microarray data dimension reduction for medical applications. In *2015 2nd International Conference on Electronics and Communication Systems (ICECS)*, pages 409–413. IEEE, 2015.

[14] Trupti M Kodinariya, Prashant R Makwana, et al. Review on determining number of cluster in k-means clustering. *International Journal*, 1(6):90–95, 2013.

[15] Michael P LaValley. Logistic regression. *Circulation*, 117(18):2395–2399, 2008.

[16] Sara C Madeira and Arlindo L Oliveira. Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM transactions on computational biology and bioinformatics*, 1(1):24–45, 2004.

[17] Sebastian Mika, Bernhard Schölkopf, Alex Smola, Klaus-Robert Müller, Matthias Scholz, and Gunnar Rätsch. Kernel pca and de-noising in feature spaces. *Advances in neural information processing systems*, 11, 1998.

[18] Fionn Murtagh and Pedro Contreras. Algorithms for hierarchical clustering: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(1):86–97, 2012.

[19] Todd G Nick and Kathleen M Campbell. Logistic regression. *Topics in biostatistics*, pages 273–301, 2007.

[20] Alan P Reynolds, Graeme Richards, Beatriz de la Iglesia, and Victor J Rayward-Smith. Clustering rules: a comparison of partitioning and hierarchical clustering algorithms. *Journal of Mathematical Modelling and Algorithms*, 5:475–504, 2006.

[21] Irina Rish et al. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46, 2001.

[22] Peter J Rousseeuw and Katrien Van Driessen. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41(3):212–223, 1999.

[23] Sajid Saif. Prostate cancer. `https://www.kaggle.com/datasets/sajidsaifi/prostate-cancer/data`, 2018. [Online; accessed February 25, 2024].

[24] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Kernel principal component analysis. In *International conference on artificial neural networks*, pages 583–588. Springer, 1997.

[25] Shan Suthaharan and Shan Suthaharan. Support vector machine. *Machine learning models and algorithms for big data classification: thinking with examples for effective learning*, pages 207–235, 2016.

[26] Eva Tuba, Ivana Strumberger, Timea Bezdan, Nebojsa Bacanin, and Milan Tuba. Classification and feature selection method for medical datasets by brain storm optimization algorithm and support vector machine. *Procedia Computer Science*, 162:307–315, 2019.

[27] Elif Derya Übeyli. Implementing automated diagnostic systems for breast cancer detection. *Expert systems with Applications*, 33(4):1054–1062, 2007.

[28] Michel Verleysen and Damien François. The curse of dimensionality in data mining and time series prediction. In *International work-conference on artificial neural networks*, pages 758–770. Springer, 2005.

[29] Geoffrey I Webb, Eamonn Keogh, and Risto Miikkulainen. Naïve bayes. *Encyclopedia of machine learning*, 15(1):713–714, 2010.

[30] World Health Organization. Cancer, 2022.

[31] Rui Xu and Donald Wunsch. Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3):645–678, 2005.

[32] Miin-Shen Yang, Chien-Yo Lai, and Chih-Ying Lin. A robust em clustering algorithm for gaussian mixture models. *Pattern Recognition*, 45(11):3950–3961, 2012.

[33] Hui Zou, Trevor Hastie, and Robert Tibshirani. Sparse principal component analysis. *Journal of computational and graphical statistics*, 15(2):265–286, 2006.

[34] Hui Zou and Lingzhou Xue. A selective overview of sparse principal component analysis. *Proceedings of the IEEE*, 106(8):1311–1320, 2018.