**Breast cancer prediction**

**(Principal Components Analysis & Classification using Discriminant Analysis)**

**Introduction**

The dataset used is from the UCI Machine Learning Repository. Principal Components Analysis (PCA) will be used as a data reduction technique. In addition, classifications will be made using Discriminant Analysis. The goal of this assignment is to create a model that can predict whether a cell nucleus is benign or malignant.

**Data set description**

The dataset, Breast Cancer Wisconsin, contains features retained from a digitized image of a fine needle aspirate (FNA) of a breast mass. These features describe the cell nucleus’ characteristics. There are 569 observations and 32 variables, from which one variable is an ID variable. The ID variable is the diagnosis revealing whether the mass is benign or malignant. This leaves 30 variables that are divided in 3 groups, where 10 different characteristics are measured differently; the mean, the standard error and the ‘worst’ or largest (mean of the three largest values). Ten real-valued features are computed for each cell nucleus:  
  
a) radius (mean of distances from center to points on the perimeter)  
b) texture (standard deviation of gray-scale values)  
c) perimeter  
d) area  
e) smoothness (local variation in radius lengths)  
f) compactness (perimeter^2 / area - 1.0)  
g) concavity (severity of concave portions of the contour)  
h) concave points (number of concave portions of the contour)  
i) symmetry  
j) fractal dimension ("coastline approximation" - 1)

**Method of analysis**

PCA will help with reducing the dimensionality of the data set without compromising the data within. PCA works by creating linear combinations given the original data. With the help of PCA I will standardize the data, calculate the eigenvectors and eigenvalues with the help of a correlation matrix, sort these in descending order and extract a suitable number of principal components. From there, the projection matrix will be constructed, and the original data set will be transformed. While PCA helps with the reduction, I will not use it to create a prediction model. The goal of this assignment is to make a model that will predict whether the extracted mass is benign or malignant, using the previously extracted principal components.

**Interpretation of results**

A quick overview of the data is presented below to begin with. Since there are numerical and categorical variables, the function **c {base}** is used, and the data is combined into vectors. The response variable is the diagnosis itself with Benign (B) or Malignant (M). It might be helpful to see how many observations are benign or malignant using the **table()** function.

B M

357 212

**>**

The mean of each of the numeric columns is presented below:

radius\_mean texture\_mean perimeter\_mean area\_mea14.13

14.13 19.29 91.97 654.89

smoothness\_mean compactness\_mean concavity\_mean concave.points\_mean

0.10 0.10 0.09 0.05

symmetry\_mean fractal\_dimension\_mean radius\_se texture\_se

0.18 0.06 0.41 1.22

perimeter\_se area\_se smoothness\_se compactness\_se

2.87 40.34 0.01 0.03

concavity\_se concave.points\_se symmetry\_se fractal\_dimension\_se

0.03 0.01 0.02 0.00

radius\_worst texture\_worst perimeter\_worst area\_worst

16.27 25.68 107.26 880.58

smoothness\_worst compactness\_worst concavity\_worst concave.points\_worst

0.13 0.25 0.27 0.11

symmetry\_worst fractal\_dimension\_worst

0.29 0.08

The standard deviations of each numeric column:

radius\_mean texture\_mean perimeter\_mean area\_mean

3.52 4.30 24.30 351.91

smoothness\_mean compactness\_mean concavity\_mean concave.points\_mean

0.01 0.05 0.08 0.04

symmetry\_mean fractal\_dimension\_mean radius\_se texture\_se

0.03 0.01 0.28 0.55

perimeter\_se area\_se smoothness\_se compactness\_se

2.02 45.49 0.00 0.02

concavity\_se concave.points\_se symmetry\_se fractal\_dimension\_se

0.03 0.01 0.01 0.00

radius\_worst texture\_worst perimeter\_worst area\_worst

4.83 6.15 33.60 569.36

smoothness\_worst compactness\_worst concavity\_worst concave.points\_worst

0.02 0.16 0.21 0.07

symmetry\_worst fractal\_dimension\_worst

0.06 0.02

Moving on, we plot the correlation between the variables. The output below shows that a many of the variables are intercorrelated.

Timeline

Description automatically generated with medium confidence

**Principal Components Analysis**

Since so many of our variables are correlated, a PCA is highly suitable for this analysis, and will help with the reduction of the data without compromising the features in the data. Below we will see how much the data can be reduced, i.e., how much of the variance in the data can be explained with only a few principal components.

Using the **prcomp()** function R runs a PCA on the data, with a limitation to 10 principal components. The data is centered and scaled (mean-corrected and standardized). From R we get the following output:

Importance of components**:**

PC1 PC2 PC3 PC4 PC5 PC6

Standard deviation 3.6444 2.3857 1.67867 1.40735 1.28403 1.09880

Proportion of Variance 0.4427 0.1897 0.09393 0.06602 0.05496 0.04025

Cumulative Proportion 0.4427 0.6324 0.72636 0.79239 0.84734 0.88759

Importance of components**:**

PC7 PC8 PC9 PC10

Standard deviation 0.82172 0.69037 0.6457 0.59219

Proportion of Variance 0.02251 0.01589 0.0139 0.01169

Cumulative Proportion 0.91010 0.92598 0.9399 0.95157

One property of PCA is that it sorts the components from largest to smallest eigenvalues, which in this output are given as standard deviations, since the data was standardized. The proportion of variance tells us how much variance each component accounts for in the data, i.e., the first principal component accounts for 44% of the total variance in the data. Cumulative proportion gives the accumulated amount of the explained variance by each component, in this case the first 10 components explain 95% of total variance.

The following step will help with determining how many principal components should be used. There are several ways to go about this, one of them being the eigenvalue > 1 rule (can only be used for standardized data), which states that the number of PCs used should correspond to the number of eigenvalues greater than one. In the output above, we can see that there are 6 such eigenvalues, for PC1 to PC6. Should this method be used, PC1 to PC6 would stand for 88% of the total variance explained.

Another way is to use a scree plot and look for the elbow in the plot. Below is a scree plot for first 15 eigenvalues, and the elbow presents itself at principal component 3. First three principal components would however only explain around 72% of the total variance, which is not low, but for the sake of having a more reliable model, I’ll set the limit at six principal components (88%).

Chart, histogram

Description automatically generated

We could also look at the cumulative variance plot below which shows that 90% of the total variance explained is set to the first six principal components.

Chart

Description automatically generated

PCA has successfully helped with reducing the dimensions of the data from 30 to 6, sacrificing only 10% of the variance.

The plot below is a plot of the first two principal components and there is an obvious clustering in the upper right side of the plot. With the original goal of the assignment in mind, we want to see how much of this cluster is because of the response variable in the data, diagnosis.

Chart, scatter chart

Description automatically generated

Highlighting the response variable, diagnosis, and plotting it shows that most of the clustering is caused by that variable, with the yellow clustering being the malignant tumor and the blue clustering being the benign tumor.

Chart, scatter chart

Description automatically generated

**Linear Discriminant Analysis**

The goal of Linear Discriminant analysis in this assignment is to find the discriminant function that will separate our data between benign and malignant points using our previous principal components. In other words, the goal is the see at which point in size, texture and concavity of the nucleus does a tumor go from being benign to malignant.

Using R, we must start by building a model using training data. A prediction is made using the test data and finally an evaluation is done using ROC, which will help by characterizing the sensitivity and specificity tradeoffs.[[1]](#footnote-1)

Firstly, we are interested in the loadings of the first six principal components we extracted earlier. These are multiplied by the scaled data, giving us principal component transformed data. Onto these transformed data, we add the diagnosis column. Using these steps, we can now define a train/test-split of the data. The data is split into 75/25, where 429 observations go into the training part, and 140 observations into the testing part. The training dataset is then used to compute the linear discriminant function. The testing dataset is then used to make a prediction. The linear discriminant function is given as:

**Conclusion**

In summary, the LDA is presented below in a ROC plot. Some points have additionally been added to the plot to make it more intuitive. These points are the “0% false-positives” (blue), “100% true-positives” (orange), “optimal cut-off” (green) and “perfect separation” (yellow). ![Chart, line chart

Description automatically generated]()

The plot shows us different options. Would we want to have a 100% true positive rate at the cost of getting some false positives? Or would we prefer 0% false positive rates at the cost of the other? This boils down to an ethical question, would we rather diagnose a healthy patient as sick (in this case having a malignant tumor), or would we rather diagnose a sick person as being healthy? None of this is easy, and because of that we can’t opt for either. This means that our optimal cut of is at the green point. The optimal point isn’t perfect, but it reduces the percentage of false positives and true positives at an equilibrium.

1. Because of the limitation of 6 pages, I will briefly explain the process in R without providing output here. Output will be provided in a separate file. [↑](#footnote-ref-1)