Assignment 2

Introduction

We have taken a tissue biopsy from an ovarian cancer patient. The biopsy includes benign and cancer cells. We measured 30 different characteristics (features) of the cells. These features represent various morphological properties as well as the expression of various proteins on the surface of the cell.

We would like to perform exploratory analysis of the measured features and see if we can design machine learning tools to separate the benign (B) and malignant (M) cells.

When you submit your assignment, please include all results and plots as well as all relevant code used for each section.

How to download the data and read the data

You can download the data named "ovarian.data" from Canvas.

You can use the following commands to read the data file:

```
ovarian.dataset <- read.delim("ovarian.data", sep=",", header = FALSE)
features <- c("perimeter", "area", "smoothness", "symmetry", "concavity",
paste("protein ", seq(1, 25) ))
names(ovarian.dataset) <- c("cell_id", "diagnosis", features) #
paste0(features,"_mean"), paste0(features,"_se"), paste0(features,"_worst"))
```

Try head(ovarian.dataset) to get a sense of the data that has been loaded.

Note 1: column 1 (cell id) represents ID of the cell, column 2 ("diagnosis") represents diagnosis (benign or malignant), and the rest of the columns represent various measured features for the cells.

Note 2: "cell id" column is for informational purposes and you will not be using this column in your analysis.

Questions

Q1. DIMENSIONALITY REDUCTION

Q1.1. Perform PCA on the features of the dataset. How much of the variation in the data is associated with PC1?

Q1.2. You want to represent 90% of the variance in the data by dimensionality reduction. How many PCs do you need to achieve this? In other word, what would be the dimensionality of the reduced feature space so that you preserve 90% of the variability in the data?

Q1.3. As you should know by now, PCA transforms the data into a new space. In a 2-D plot, can you plot the observations corresponding to the first two important PCs? Note, use two different colors to represent the two classes of cells.

Q1.4. Can you plot the "area" and "concavity" features associated with the cells?

Q1.5. What is the difference between the two plots? Which one gives you better separation between the classes and why?

Q1.6. Plot the distribution of the PCs. *Hint:* you can use boxplot on the transformed dataset.

Hint 1: when doing PCA, make sure you set scale and center arguments to TRUE.

Hint 2: try the summary() function on the PCA results.

Hint 3: PCA transforms the data into new dimension.

Q2. CLUSTERING

When comparing model predictions to true labels, obtain a confusion matrix and include this result in your submission. You can obtain this by using the table() function like so:
table(predictions, labels)

Q2.1. Apply kmeans clustering on the data and identify two clusters within your dataset. What is the concordance between the clusters that you have identified and the true labels of the cells (Benign vs Malignant).

Hint: From clustering, you get the cluster membership of each cell. Note, you need to identify two clusters. From there, you can explore the relationship between the clusters and the true labels. You can then report accuracy, precision and recall of the methods.

Hint: Use ifelse(cluster == 1, "M", "B") to convert the clusters to diagnosis labels. Remember, there is a 50-50 chance that cluster 1 corresponds to the benign label.

Hint: Don't forget to scale the data beforehand.

Q2.2. Repeat the kmeans analysis 10 times and report the mean accuracy across the 10 runs. Why are the results different in each run?

Q2.3. Repeat the same analysis but with the top 5 PCs.
Hint: From the PCA results, take the top 5 PCs and repeat the analysis. Do not scale the PCA data, however, as PCA has already taken care of this.

Q2.4. Compare the results between Q2.2. and Q2.3.
Hint: Do the results get better or worse? Why?

Q3. CLASSIFICATION

Divide your data into training and test sets using the following command:

```
ovarian.dataset.train <- ovarian.dataset[sample(nrow(ovarian.dataset))[1:
(nrow(ovarian.dataset)/2)],]
ovarian.dataset.test <- ovarian.dataset[sample(nrow(ovarian.dataset))
[(nrow(ovarian.dataset)/2):(nrow(ovarian.dataset))],]
```

Q3.1. Design a logistic regression classifier to identify (differentiate) benign and malignant cells. Report the performance of the classification technique on the training and test sets. You can report accuracy, precision and recall. Compare the performance of the classifier on the training and test set and provide a reason as to why one is better than the other.

Hint: Use the "binomial" option while training and use predict(..., type="response") to obtain the result as a probability. Then use P=0.5 as the threshold to separate probabilities into "M" and "B".

Hint: Do not worry if the model does not converge.

Q3.2. Repeat the same task as Q3.1. with the top 5 PCs.

Q3.3. Compare the results between Q3.1. and Q3.2. Do the results get better or worse? Why?

Q3.4. Compare the results of the clustering and classification methods. Which one gives you better result?

Q3.5. Install the library "ROCR" and load it. Then, run the following lines using your trained logistic regression model:

```
pred.prob <- predict(your.model, ovarian.dataset, type="response")
predict <- prediction(pred.prob, ovarian.dataset$diagnosis,
label.ordering=c("B","M"))
perform <- performance(predict,"tpr","fpr")
plot(perform,colorize=TRUE)
```

This will generate an ROC curve of the performance of the classifier for a range of thresholds. Take a look at https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5 for a summary of ROC curves.

Given our ROC curve, what would you say it tells us about the overlap of the two classes? What can we say about the model's separability? How does an ROC curve tell us more about the model's performance than a single sensitivity/specificity measure?

Q3.6. Design another classifier (using a different classification method) and repeat Q3.1-3.

List of functions to use:

Note: This list is not comprehensive, but includes most of the functions you will need.

library(ggplot2), library(ROCR), prcomp(), summary(), as.data.frame(), ggplot(), boxplot(), scale(), kmeans(), ifelse(), table(), glm(), predict()