Datamining Assignment 3

Elkhan Ismayilzada

May 13, 2021

Exercise 1

1. **Principal Component Analysis (PCA)** is an unsupervised machine learning approach for dimensionality reduction. The basic theory behind principal component analysis (PCA) is to minimize the dimensionality of a dataset consisting of several variables that are associated with each other, either strongly or loosely, while maintaining as much variance as possible in the dataset.In order to achieve this, the variables of the dataset are converted into a new collection of variables which consists of mixture of variables or attributes from initial dataset with the goal of retaining as much variance as possible. This new collection of variables is known as Principal Components (PCs) and the variable or component with maximum variance is called as Dominant Principal Component. The components are sorted according to their retention of variance in descending order.

Linear Discriminant Analysis (LDA) is a supervised machine learning method that aims to differentiate two groups or classes. The basic theory behind LDA is to maximize the difference between two classes or groups so as to make the best classification decision.In order to achieve this, firstly, it creates a new linear axis and projects the data points on that axis. Afterwards, it finds out the best line that can distinguish the classes or groups.

Although both aims for **dimensionality reduction**, PCA is unsupervised machine learning method whereas LDA is supervised machine learning method. In PCA, we create new axis that will maximize the variance but in LDA, we create new axis that will maximize separation between classes. In a more detailed manner, LDA does not search for the principal component; instead, it examines which form of features provides the best differentiation to distinguish the data.

2. **A decision tree** is a supervised machine learning method that is used for both classification and regression. The basic theory behind Decision Tree is to construct a tree that goes from observations about an item to conclusions about the item's target value. The goal of decision tree is to predict the value of the target variable based on conditional statements. As an example consider following the Figure 1. Here we try to predict the commute time. For example, if we leave at 8 am, we predicted that the commute time will be long.

Random forest is a tree-based machine learning algorithm that makes decisions by combining the influence of several decision trees. The reason why it is called as "random forest" is because it consists of many randomly generated decision trees. To put in simple words, random forest algorithm takes in outputs of decision trees and generates the final output by averaging. This process of combining the output of individual models is called Ensemble Learning so we can

say that Random forest is ensemble learning method for classification, regression, etc.

The difference between Random forest and Decision tree is how they generate the output and how fast it is made. In Decision tree, output is made by combining some decisions. On the other hand, in Random forest, output is generated by combining the outputs of many decision trees. Therefore, in terms of the speed, Random forest is much slower than the Decision tree but the output is more reliable.

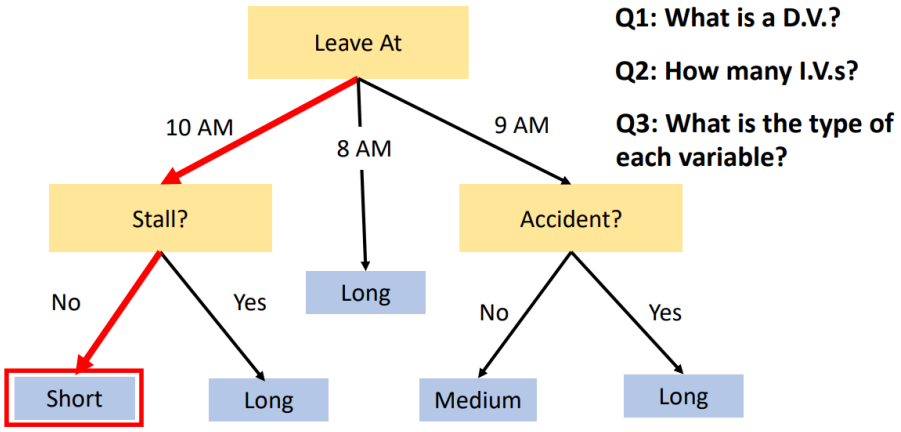

Figure 1: Decision tree

Exercise 2

2.1

We can convert the table as in Figure 2. The accuracy is then

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{2.1}$$

From accuracy we get the information about how much out of all classes did we predict it correctly.
The equation of sensitivity (recall) is

$$sensitivity = \frac{TP}{TP + FN} \tag{2.2}$$

Sensitivity gives information about how much out of all positive classes (Disorder in this case) did we predict it correctly.
Specificity, on the other hand, is

$$specificity = \frac{TN}{TN + FP} \tag{2.3}$$

It gives information about how much out of all negative classes (No disorder in this case) did we predict it correctly

Now we can easily find out all from the tables
For table 1

$$\begin{aligned} accuracy &= \frac{8+929}{8+45+929+18} \cdot 100 = 93.7\% \\ sensitivity &= \frac{8}{8+18} \cdot 100 = 30.77\% \\ specificity &= \frac{929}{929+45} \cdot 100 = 95.38\% \end{aligned} \tag{2.4}$$

And for table 2

$$\begin{aligned} accuracy &= \frac{12+914}{12+60+914+14} \cdot 100 = 92.6\% \\ sensitivity &= \frac{12}{12+14} \cdot 100 = 46.15\% \\ specificity &= \frac{914}{914+60} \cdot 100 = 93.84\% \end{aligned} \tag{2.5}$$

2.2
With naive rule, TP and FP are zero, hence the accuracy is

$$accuracy = \frac{0+974}{0+0+974+26} \cdot 100 = 97.4\% \tag{2.6}$$

It is more than the accuracy obtained in Logistic Regression and Decision Tree.
2.3
I would prefer to use Decision Tree in terms of accuracy, sensitivity and specificity. The reason is because sensitivity is crucial in this dataset. If we consider a patient as "no disorder" although he is "disorder", this is the worst mistake we could do because it may cost us the life of the patient.
2.4 In order to improve accuracy, I would use Random Forest Classifier. By using Random Forest, we will be able to reduce the possibility of overfitting due to imbalanced data.

		Predicted class	
		Yes	No
Actual class	Yes	TP: True positive	FN: False negative
	No	FP: False positive	TN: True negative

Figure 2: Confusion matrix

Exercise 3

3.1

Classification criterion	Sensitivity	1-Specificity
< 24	1	1
< 35	1	0.75
< 37	1	0.5
< 42	0.8333	0.5
< 49	0.8333	0.25
< 54	0.6667	0.25
< 56	0.5	0.25
< 68	0.5	0
< 72	0.3333	0
< 73	0.1667	0

3.2

ROC curve shows tradeoff between true positives and false positives over the threshold (probability cutoff). If the curve is closer to the top-left, it indicates better performance. On x axis we have False Positive Rate (1-specificity) and on y axis we have True Positive Rate (sensitivity). Here is the plot of ROC curve and implementation.

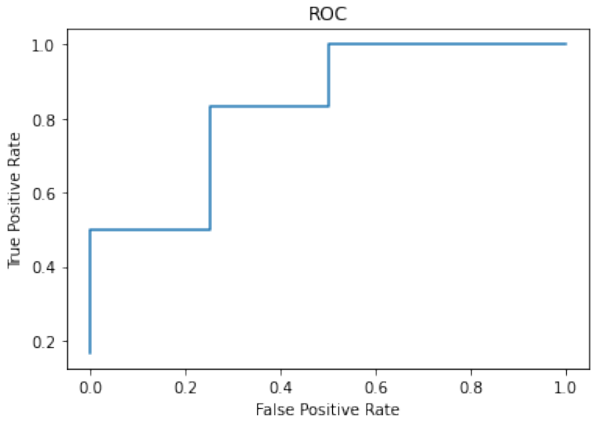

Figure 3: ROC

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
```

```
df = pd.DataFrame({"True Positive Rate":[1,1,1,0.8333,0.8333,0.6667,0.5,0.5,0.3333,0.1667],
                   "False Positive Rate":[1,0.75,0.5,0.5,0.25,0.25,0.25,0,0,0]})
plt.plot(df["False Positive Rate"], df["True Positive Rate"]);
plt.savefig("my_plot.png")
plt.title("ROC");
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate");
```

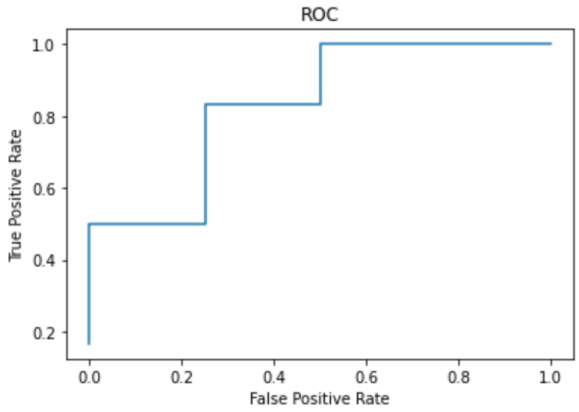

Figure 4: Implementation

Exercise 4

4.1

$$\mu_0 = \begin{bmatrix} \frac{2.58+2.16+3.27}{3} & \frac{4.46+6.22+3.52}{3} \end{bmatrix} = \begin{bmatrix} 2.67 & 4.7333 \end{bmatrix}$$
$$\mu_1 = \begin{bmatrix} \frac{2.84+2.54+3.54+3.18}{4} & \frac{6.57+7.83+5.82+5.71}{4} \end{bmatrix} = \begin{bmatrix} 3.025 & 6.4825 \end{bmatrix} \tag{4.1}$$

Scatter matrices are given by following equation

$$S_i = \sum_{x \in \omega_i} (x - \mu_i)(x - \mu_i)^T \tag{4.2}$$

Therefore

$$\begin{aligned} S_0 = &\left(\begin{bmatrix}2.58\\4.46\end{bmatrix} - \begin{bmatrix}2.67\\4.7333\end{bmatrix}\right) \cdot \left(\begin{bmatrix}2.58 & 4.46\end{bmatrix} - \begin{bmatrix}2.67 & 4.7333\end{bmatrix}\right) \\ &+\left(\begin{bmatrix}2.16\\6.22\end{bmatrix} - \begin{bmatrix}2.67\\4.7333\end{bmatrix}\right) \cdot \left(\begin{bmatrix}2.16 & 6.22\end{bmatrix} - \begin{bmatrix}2.67 & 4.7333\end{bmatrix}\right) \\ &+\left(\begin{bmatrix}3.27\\3.52\end{bmatrix} - \begin{bmatrix}2.67\\4.7333\end{bmatrix}\right) \cdot \left(\begin{bmatrix}3.27 & 3.52\end{bmatrix} - \begin{bmatrix}2.67 & 4.7333\end{bmatrix}\right) = \begin{bmatrix}0.6282 & -1.4616\\-1.4616 & 3.7571\end{bmatrix} \end{aligned}$$

$$\begin{aligned} S_1 = &\left(\begin{bmatrix}2.84\\6.57\end{bmatrix} - \begin{bmatrix}3.025\\6.4825\end{bmatrix}\right) \cdot \left(\begin{bmatrix}2.84 & 6.57\end{bmatrix} - \begin{bmatrix}3.025 & 6.4825\end{bmatrix}\right) \\ &+\left(\begin{bmatrix}2.54\\7.83\end{bmatrix} - \begin{bmatrix}3.025\\6.4825\end{bmatrix}\right) \cdot \left(\begin{bmatrix}2.54 & 7.83\end{bmatrix} - \begin{bmatrix}3.025 & 6.4825\end{bmatrix}\right) \\ &+\left(\begin{bmatrix}3.54\\5.82\end{bmatrix} - \begin{bmatrix}3.025\\6.4825\end{bmatrix}\right) \cdot \left(\begin{bmatrix}3.54 & 5.82\end{bmatrix} - \begin{bmatrix}3.025 & 6.4825\end{bmatrix}\right) \\ &+\left(\begin{bmatrix}3.18\\5.71\end{bmatrix} - \begin{bmatrix}3.025\\6.4825\end{bmatrix}\right) \cdot \left(\begin{bmatrix}3.18 & 5.71\end{bmatrix} - \begin{bmatrix}3.025 & 6.4825\end{bmatrix}\right) = \begin{bmatrix}0.5587 & -1.1307\\-1.1307 & 2.8591\end{bmatrix} \end{aligned} \tag{4.3}$$

4.2

Within-class scatter matrix is the sum of S_0 and S_1

$$S_W = \begin{bmatrix}0.6282 + 0.5587 & -1.4616 - 1.1307\\-1.4616 - 1.1307 & 3.7571 + 2.8591\end{bmatrix} = \begin{bmatrix}1.1869 & -2.5922\\-2.5922 & 6.6161\end{bmatrix} \tag{4.4}$$

Between-class scatter matrix is given by

$$S_B = \sum_{i=0}^{c} N_i (m_i - m)(m_i - m)^T \tag{4.5}$$

Where N_i is the sample size for class i, m is the overall mean, m_i is the mean for class i. The m is

$$m = \begin{bmatrix}\frac{2.58+2.16+3.27+2.84+2.54+3.54+3.18}{7} & \frac{4.46+6.22+3.52+6.57+7.83+5.82+5.71}{7}\end{bmatrix} = \begin{bmatrix}2.8729 & 5.7329\end{bmatrix} \tag{4.6}$$

Now we can find S_B

$$\begin{aligned} S_B = 3 \cdot &\begin{bmatrix}2.67\\4.7333\end{bmatrix} - \begin{bmatrix}2.8729\\5.7329\end{bmatrix} \cdot \begin{bmatrix}2.67 & 4.7333\end{bmatrix} - \begin{bmatrix}2.8729 & 5.7329\end{bmatrix} + \\ 4 \cdot &\begin{bmatrix}3.025\\6.4825\end{bmatrix} - \begin{bmatrix}2.8729\\5.7329\end{bmatrix} \cdot \begin{bmatrix}3.025 & 6.4825\end{bmatrix} - \begin{bmatrix}2.8729 & 5.7329\end{bmatrix} = \begin{bmatrix}0.216 & 1.0645\\1.0645 & 5.245\end{bmatrix} \end{aligned} \tag{4.7}$$

4.3

The optimal w^* is given by

$$w^* = S_W^{-1} \cdot (\mu_0 - \mu_1) \tag{4.8}$$

Therefore

$$w^* = \begin{bmatrix}1.1869 & -2.5922\\-2.5922 & 6.6161\end{bmatrix}^{-1} \cdot \left(\begin{bmatrix}2.67\\4.7333\end{bmatrix} - \begin{bmatrix}3.025\\6.4825\end{bmatrix}\right) = \begin{bmatrix}-6.0754\\-2.6447\end{bmatrix} \tag{4.9}$$

Exercise 5

5.1

The equation for the entropy is

$$Entropy = -\sum_{i=1}^{k} \frac{n_i}{n} \log \frac{n_i}{n} \tag{5.1}$$

And the gain split is the change in entropy after splitting.

$$GAIN_{split} = Entropy(p) - \sum_{i=1}^{k} \frac{n_i}{n} Entropy(i) \tag{5.2}$$

Where p is the parent node The Gain Ratio is

$$GainRATIO_{split} = \frac{GAIN_{split}}{SplitINFO} \tag{5.3}$$

Where $SplitINFO$ is

$$SplitINFO = -\sum_{i=1}^{k} \frac{n_i}{n} \log \frac{n_i}{n} \tag{5.4}$$

The entropy of the whole data is

$$Entropy(data) = -\frac{9}{14} \log \frac{9}{14} - \frac{5}{14} \log \frac{5}{14} = 0.9403 \tag{5.5}$$

We start with Age attribute

$$\begin{aligned}
age(youth) &= [yes, no, no, no, yes] \\
age(middle-aged) &= [yes, yes, yes, yes] \\
age(senior) &= [yes, no, no, yes, yes] \\
Entropy(youth) &= -\frac{2}{5} \log \frac{2}{5} - \frac{3}{5} \log \frac{3}{5} = 0.9709 \\
Entropy(middle-aged) &= -\frac{4}{4} \log \frac{4}{4} = 0 \\
Entropy(senior) &= -\frac{3}{5} \log \frac{3}{5} - \frac{2}{5} \log \frac{2}{5} = 0.9709 \\
Entropy(age) &= \frac{5}{14} \cdot 0.9709 + 0 + \frac{5}{14} \cdot 0.9709 = 0.6935 \\
GAIN_{split}(age) &= Entropy(data) - Entropy(age) = 0.9403 - 0.6935 = 0.2468 \\
SplitINFO(age) &= -\frac{5}{14} \cdot \log \frac{5}{14} - \frac{4}{14} \log \frac{4}{14} - \frac{5}{14} \log \frac{5}{14} = 1.5774 \\
GainRATIO_{split}(age) &= \frac{0.2468}{1.5774} = 0.1565
\end{aligned} \tag{5.6}$$

Now we will do the same procedure with Health Concern attribute

$$
\begin{aligned}
concern(low) &= [yes, yes, no, yes] \\
concern(medium) &= [yes, yes, no, no, yes, yes] \\
concern(high) &= [yes, no, yes, no] \\
Entropy(low) &= -\frac{3}{4}\log\frac{3}{4} - \frac{1}{4}\log\frac{1}{4} = 0.8113 \\
Entropy(medium) &= -\frac{4}{6}\log\frac{4}{6} - \frac{2}{6}\log\frac{2}{6} = 0.9183 \\
Entropy(high) &= -\frac{2}{4}\log\frac{2}{4} - \frac{2}{4}\log\frac{2}{4} = 1 \\
Entropy(concern) &= \frac{4}{14}\cdot 0.8113 + \frac{6}{14}\cdot 0.9183 + \frac{4}{14}\cdot 1 = 0.9111 \\
GAIN_{split}(concern) &= Entropy(data) - Entropy(concern) = 0.9403 - 0.9111 = 0.0292 \\
SplitINFO(concern) &= -\frac{4}{14}\cdot\log\frac{4}{14} - \frac{6}{14}\log\frac{6}{14} - \frac{4}{14}\log\frac{4}{14} = 1.5567 \\
GainRATIO_{split}(concern) &= \frac{0.0292}{1.5567} = 0.0188
\end{aligned}
\tag{5.7}
$$

Next is the Exercise attribute

$$
\begin{aligned}
exercise(seldom) &= [yes, yes, no, no, no, no, yes] \\
exercise(frequent) &= [yes, yes, yes, yes, no, yes, yes] \\
Entropy(seldom) &= -\frac{3}{7}\log\frac{3}{7} - \frac{4}{7}\log\frac{4}{7} = 0.9852 \\
Entropy(frequent) &= -\frac{6}{7}\log\frac{6}{7} - \frac{1}{7}\log\frac{1}{7} = 0.5917 \\
Entropy(exercise) &= \frac{7}{14}\cdot 0.9852 + \frac{7}{14}\cdot 0.5917 = 0.7885 \\
GAIN_{split}(exercise) &= Entropy(data) - Entropy(exercise) = 0.9403 - 0.7885 = 0.1518 \\
SplitINFO(exercise) &= -\frac{7}{14}\cdot\log\frac{7}{14} - \frac{7}{14}\log\frac{7}{14} = 1 \\
GainRATIO_{split}(exercise) &= \frac{0.1518}{1} = 0.1518
\end{aligned}
\tag{5.8}
$$

Last is the Health Status attribute

$$
\begin{aligned}
status(fair) &= [yes, yes, no, yes, no, yes, yes, yes] \\
status(excellent) &= [yes, yes, no, yes, no, no] \\
Entropy(fair) &= -\frac{6}{8}\log\frac{6}{8} - \frac{2}{8}\log\frac{2}{8} = 0.8113 \\
Entropy(excellent) &= -\frac{3}{6}\log\frac{3}{6} - \frac{3}{6}\log\frac{3}{6} = 1 \\
Entropy(status) &= \frac{8}{14}\cdot 0.8113 + \frac{6}{14}\cdot 1 = 0.8922 \\
GAIN_{split}(status) &= Entropy(data) - Entropy(status) = 0.9403 - 0.8922 = 0.0481 \\
SplitINFO(status) &= -\frac{8}{14}\cdot\log\frac{8}{14} - \frac{6}{14}\log\frac{6}{14} = 0.9852 \\
GainRATIO_{split}(status) &= \frac{0.0481}{0.9852} = 0.0488
\end{aligned}
\tag{5.9}
$$

The splitting criterion should be with Age attribute since the Gain Ratio is highest (a bit higher than Exercise)

5.2

After splitting with Age attribute we have 3 nodes

$$\begin{aligned} age(youth) &= [yes, no, no, no, yes] \\ age(middle-aged) &= [yes, yes, yes, yes] \\ age(senior) &= [yes, no, no, yes, yes] \end{aligned} \tag{5.10}$$

Classification error at node t is given by

$$Error(t) = 1 - max_i P(i|t) \tag{5.11}$$

As a result

$$\begin{aligned} Error(youth) &= 1 - \frac{3}{5} = \frac{2}{5} \\ Error(middle-aged) &= 1 - \frac{4}{4} = 0 \\ Error(senior) &= 1 - \frac{3}{5} = \frac{2}{5} \\ Error(root) &= 1 - \frac{9}{14} = \frac{5}{14} \end{aligned} \tag{5.12}$$

References

Aman Kapri. (2020, February 17). PCA vs LDA vs T-SNE — Let's Understand the difference between them! Retrieved May 6, 2021, from Medium website: https://medium.com/analytics-vidhya/pca-vs-lda-vs-t-sne-lets-understand-the-difference-between-them-22fa6b9be9d0