Assignment 2b

David Boerema (s3683869)
Marios Souroulla (s4765125)
Marius Captari (s4865928)
Max Valk (s3246922)

Group 3

September 28, 2021

B1.1

a)

	predicted positive	predicted negative
actual positive	15	20
actual negative	0	1965

True Positive (TP) = 15
False Negative (FN) = 20
False Positive (FP) = 0
True Negative (TN) = 1965

$$\text{True Positive Rate} = \frac{TP}{TP+FN} = \frac{15}{15+20} \approx 0.428$$
$$\text{True Negative Rate} = \frac{TN}{TN+FP} = \frac{1965}{1965+0} = 1$$
$$\text{Precision} = \frac{TP}{TP+FP} = \frac{15}{15+0} = 1$$
$$\text{Recall} = \frac{TP}{TP+FN} = \frac{15}{15+20} \approx 0.428$$
$$\text{Specificity} = \frac{TN}{TN+FP} = \frac{1965}{1965+0} = 1$$
$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} = \frac{15+1965}{15+20+0+1965} = 0.99$$
$$\text{F1 score} = \frac{2 \cdot TP}{2 \cdot TP+FP+FN} = \frac{2 \cdot 15}{2 \cdot 15+0+20} = 0.6$$
$$\text{Cohen's Kappa} = \frac{2 \cdot (TP \cdot TN - FN \cdot FP)}{(TP+FP) \cdot (FP+TN) + (TP+FN) \cdot (FN+TN)} = 0.6$$

b)

$$\text{Imbalance ratio} = \frac{\text{Size of minority class}}{\text{Size of majority class}} = \frac{35}{1965} \approx 0.0178$$

c)

Given that on this particular problem there is a high class imbalance, a metric such as accuracy wouldn't be appropriate since it would reward a model that such predicts the majority class. Recall in this case for example, gives us a better idea of how well our model is correctly predicting positive samples, since it doesn't take into account the class imbalance and we are just looking at the minority class. A good metric for this example would be $\mathbf{F_1}$**Score** since it combines both accuracy and precision and it seeks to balance both concerns of unbalanced data. Another possible options is **Cohen's Kappa** since it compares observed accuracy with random chance accuracy, measuring how closely the predicted instances match the actual data while controlling for the accuracy of a random classifier as measured by the expected accuracy.

B1.2

To determine the information gain for each feature w.r.t. determining whether the cyclist went on a ride or not, we use equation 1.

$$IG = H_{old} - H_{new} \tag{1}$$

This equation determines the information gain based on the entropy before including the feature (old) and after including the feature (new). We do this for every feature in Table 1.
First, we calculate H_{old}. For this we use the total amount of rows (14), the frequency of "Yes" (9) and the frequency of "No" (4):

$$\begin{aligned} H_{old} &= -\left(\frac{9}{14}log_2(\frac{9}{14}) + \frac{5}{14}log_2(\frac{5}{14})\right) \\ &\approx 0.94 \end{aligned}$$

This shows that we have pretty high entropy. Next, we take every feature, find the total amount and amount of occurrences of "Yes" and "No" per feature category and use them to calculate H_{new}. We then plug the result along with H_{old} into equation 1 to get the information gain for the feature.

Sky condition

Sunny: (2 Yes, 3 No); Cloudy: (4 Yes, 0 No); Rain: (3 Yes, 2 No)

$$\begin{aligned} H_{new} &= \frac{5}{14}H_{sunny} + \frac{4}{14}H_{cloudy} + \frac{5}{14}H_{rain} \\ H_{sunny} &= -\left(\frac{2}{5}log_2(\frac{2}{5}) + \frac{3}{5}log_2(\frac{3}{5})\right) \\ H_{cloudy} &= -\left(1log_2(1)\right) = 0 \\ H_{rain} &= -\left(\frac{3}{5}log_2(\frac{3}{5}) + \frac{2}{5}log_2(\frac{2}{5})\right) \\ H_{new} &= -\frac{5}{14}\left(\frac{2}{5}log_2(\frac{2}{5}) + \frac{3}{5}log_2(\frac{3}{5})\right) + 0 - \frac{5}{14}\left(\frac{3}{5}log_2(\frac{3}{5}) + \frac{2}{5}log_2(\frac{2}{5})\right) \\ IG &\approx 0.247 \end{aligned}$$

Temperature

High: (2 Yes, 2 No); Medium: (4 Yes, 2 No); Low: (3 Yes, 1 No)

$$
\begin{aligned}
H_{new} &= \frac{4}{14}H_{high} + \frac{6}{14}H_{medium} + \frac{4}{14}H_{low} \\
H_{high} &= -\left(\frac{2}{4}log_2(\frac{2}{4}) + \frac{2}{4}log_2(\frac{2}{4})\right) = 1 \\
H_{medium} &= -\left(\frac{4}{6}log_2(\frac{4}{6}) + \frac{2}{6}log_2(\frac{2}{6})\right) \\
H_{low} &= -\left(\frac{3}{4}log_2(\frac{3}{4}) + \frac{1}{4}log_2(\frac{1}{4})\right) \\
H_{new} &= \frac{4}{14} - \frac{6}{14}\left(\frac{4}{6}log_2(\frac{4}{6}) + \frac{2}{6}log_2(\frac{2}{6})\right) - \frac{4}{14}\left(\frac{3}{4}log_2(\frac{3}{4}) + \frac{1}{4}log_2(\frac{1}{4})\right) \\
IG &\approx 0.029
\end{aligned}
$$

Humidity

High: (3 Yes, 4 No); Low: (6 Yes, 1 No)

$$
\begin{aligned}
H_{new} &= \frac{7}{14}H_{high} + \frac{7}{14}H_{low} \\
H_{high} &= -\left(\frac{3}{7}log_2(\frac{3}{7}) + \frac{4}{7}log_2(\frac{4}{7})\right) \\
H_{low} &= -\left(\frac{6}{7}log_2(\frac{6}{7}) + \frac{1}{7}log_2(\frac{1}{7})\right) \\
H_{new} &= -\frac{7}{14}\left(\frac{3}{7}log_2(\frac{3}{7}) + \frac{4}{7}log_2(\frac{4}{7})\right) - \frac{7}{14}\left(\frac{6}{7}log_2(\frac{6}{7}) + \frac{1}{7}log_2(\frac{1}{7})\right) \\
IG &\approx 0.152
\end{aligned}
$$

Windy

True: (3 Yes, 3 No); False: (6 Yes, 2 No)

$$
\begin{aligned}
H_{new} &= \frac{6}{14}H_{true} + \frac{8}{14}H_{false} \\
H_{true} &= -\left(\frac{3}{6}log_2(\frac{3}{6}) + \frac{3}{6}log_2(\frac{3}{6})\right) = 1 \\
H_{false} &= -\left(\frac{6}{8}log_2(\frac{6}{8}) + \frac{2}{8}log_2(\frac{2}{8})\right) \\
H_{new} &= \frac{6}{14} - \frac{8}{14}\left(\frac{6}{8}log_2(\frac{6}{8}) + \frac{2}{8}log_2(\frac{2}{8})\right) \\
IG &\approx 0.048
\end{aligned}
$$

Results

Now that the information gain has been calculated, the features can be ranked. A higher information gain is better, as this results in less entropy. We get the following feature ranking:

1. Sky condition ≈ 0.247
2. Humidity ≈ 0.152
3. Windy ≈ 0.048
4. Temperature ≈ 0.029

B2.1

The goal of this exercise was to implement a Matlab function to perform simple imputations. This function gets as input an input Matrix $X_{missing}$ of size n x p which has missing values (nan), a vector S of size 1 x p

which specifies the type of each feature, and returns a matrix X_{full} of the same size with the missing values being replaced by the mode (if the feature is categorical) or the mean (if the feature is continuous).

The implementation is straightforward. We loop through all the columns (features) of the input matrix and for each column we calculate the replace value. The replace value is either the mode (for categorical features) or the mean (for continuous data), based on the input vector S (for this implementation we use S_i=0 for continuous features and S_i=1 for categorical features). When calculating the replace value we have to omit the NaN values. Then, we replace the NaN values of each feature with the corresponding replace value we have calculated earlier.

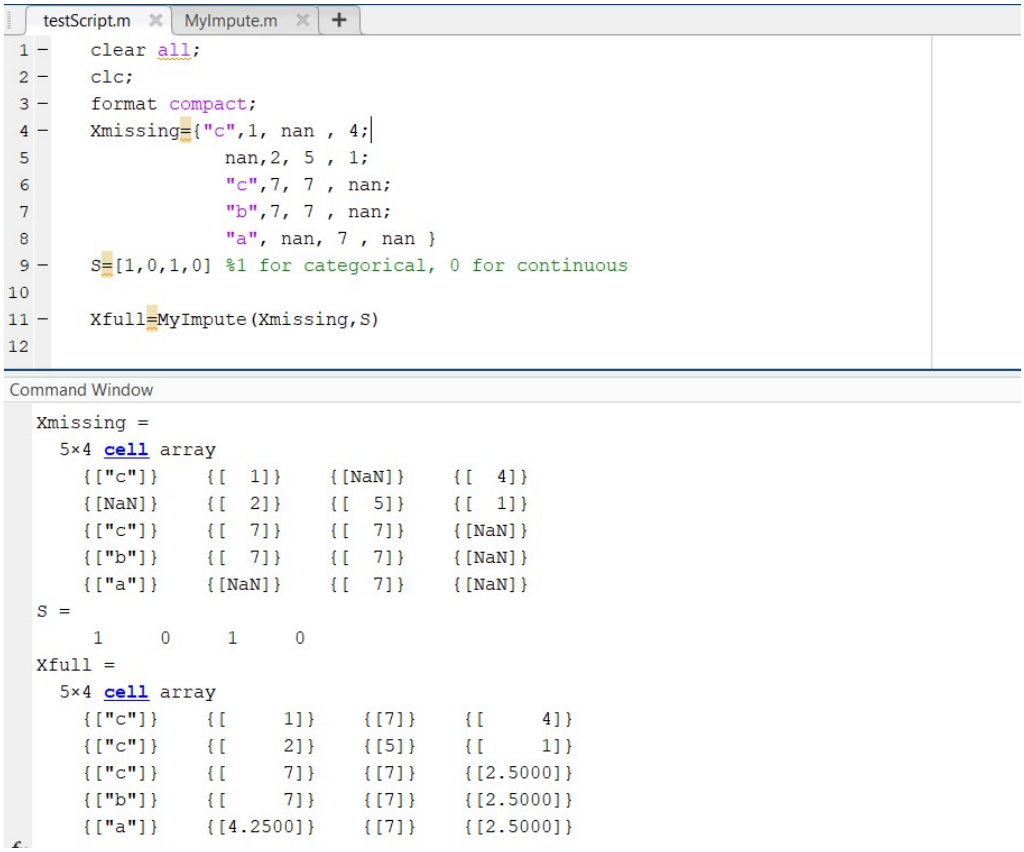

Figure 1: Example of simple imputation

Figure 1 shows the output of the function myImpute for some sample data.

B2.2

The implementation of the relief function can be found in the B2.2 folder. A test script is provided, MyReliefDemo.m, that computes the relief function for table 1 in the assignment. The weights obtained using this approach can be seen in Table 1.

	Sky condition	Temperature	Humidity	Windy
Relief	0.2857	0	0.0714	0.1429
IG	0.247	0.029	0.152	0.048

Table 1: A comparison between weights obtained from the relief algorithm with the information gain, on the cycling dataset.

Both methods agree on ranking the sky condition as the most relevant feature, and temperature as the least, but disagree on the position of the windiness and humidity. As computing feature relevance using information gain does not take interactions between features on class outcome into account, whereas relief does, this indicates that there likely is some interaction between windy and another feature such that the combination is more predictive. This intuition has been confirmed by creating new features that represent the interaction between the feature windy, and the others, by creating a new column for each (Sky_x_Windy, Temperature_x_Windy, and Humidity_x_Windy). Each entry in each column is then filled with the appended values of the two features relevant to that column. For example, the first entry in Sky_x_Windy is "SunnyFalse". If we then run the

relief algorithm again (including both old and new features), the feature that has the highest relevance is the interaction feature between the sky and wind condition (see Table 2).

	Sky_x_Windy	Temperature_x_Windy	Humidity_x_Windy
Relief	1.0000	0.2857	0.0000

Table 2: The relief weights of the new interaction variables.

If we examine these two features in the given data set, we see that they are sufficient for classifying all data points, except if the sky condition is sunny (see Table 3).

	Sunny	Cloudy	Rainy
True	Yes/No	Yes	No
False	Yes/No	Yes	Yes

Table 3: The classification outcomes by only looking at the windiness (rows) and sky condition (columns).