Introduction to Data Science: Assignment 3

Eelke Landsaat (s4056868)
Jesse Reurink (s3771385)
Darragh Spillane (s5270855)
Darren Zammit (s5284236)

Group 10

September 28, 2022

3.1.1

Confusion matrix for the classification of fraudulent transactions:

		predicted	
		positive	negative
true	positive	15	20
	negative	0	1965

a)

$$\begin{aligned}
\text{Accuracy} &= \frac{TP+TN}{TP+FP+TN+FN} = \frac{15+1965}{15+0+1965+20} = 0.99 \\
\text{Recall} &= \frac{TP}{TP+FN} = \frac{15}{15+20} = 0.42857 \\
\text{Precision} &= \frac{TP}{TP+FP} = \frac{15}{15+0} = 1 \\
\text{Specificity} &= \frac{TN}{FP+TN} = \frac{1965}{0+1965} = 1 \\
\text{F1-score} &= \frac{2}{\frac{1}{\text{precision}}+\frac{1}{\text{recall}}} = \frac{2}{\frac{1}{1}+\frac{1}{0.42857}} = 0.5999
\end{aligned}$$

b) The majority class in this case is the negative class, as it has 1965 instances compared to the 15 + 20 = 35 instances in the positive class. The imbalance ratio is therefore calculated by:

$$\text{IR} = \frac{1965}{35} = 56.1429$$

c) Since the data regards fraudulent transactions, false positives are not as bad as false negatives (we do not want to miss any fraud, but blaming someone for fraud when they are not fraudulent can later be resolved). The best performance metric should thus give an idea of the number of false negative classifications. The best candidate for this is the recall: the higher the recall, the fewer false negatives there are compared to the number of true positives.
Additionally, the F1-score would pose as a second best metric for the job, as it also takes the number of false positives into account (although not as bad as false negatives, these are still the cause of some stressful and inconvenient situations for innocent people). A flaw is that the F1-score is the harmonic mean of the precision and the recall, meaning that it weighs both measures with equal importance, when in fact we care more about the recall than about the precision.

3.1.2

Factors for determining whether or not to cycle:

Sky condition	Temperature	Humidity	Windy	Cycle
Sunny	High	High	False	No
Sunny	High	High	True	No
Cloudy	High	High	False	Yes
Rain	Mid	High	False	Yes
Rain	Low	Low	False	Yes
Rain	Low	Low	True	No
Cloudy	Low	Low	True	Yes
Sunny	Mid	High	False	No
Sunny	Low	Low	False	Yes
Rain	Mid	Low	False	Yes
Sunny	Mid	Low	True	Yes
Cloudy	Mid	High	True	Yes
Cloudy	High	Low	False	Yes
Rain	Mid	High	True	No

For the feature "sky condition" we can start to calculate the information gain I by counting the number of occurrences of each value, as well as the total number of samples.

Sunny = 5
Cloudy = 4
Rain = 5
Total = 14

We can then continue with the calculation of the entropy $H_{old,sky}$ without taking into account the prediction of the output variable.

$$H_{old,sky} = -\frac{5}{14}\log_2(\frac{5}{14}) - \frac{4}{14}\log_2(\frac{4}{14}) - \frac{5}{14}\log_2(\frac{5}{14}) = 1.57741$$

In the same way, we can calculate $H_{old,temp}$, $H_{old,hum}$ and $H_{old,wind}$:

$$H_{old,temp} = -\frac{4}{14}\log_2(\frac{4}{14}) - \frac{6}{14}\log_2(\frac{6}{14}) - \frac{4}{14}\log_2(\frac{4}{14}) = 1.55666$$

$$H_{old,hum} = -\frac{7}{14}\log_2(\frac{7}{14}) - \frac{7}{14}\log_2(\frac{7}{14}) = 1$$

$$H_{old,wind} = -\frac{6}{14}\log_2(\frac{6}{14}) - \frac{8}{14}\log_2(\frac{8}{14}) = 0.98523$$

To calculate I for each feature, we will also need H_{new} for each feature. To this end, we create a new table, which we order by whether the cycle column indicates "yes" or "no":

Sky condition	Temperature	Humidity	Windy	Cycle
Cloudy	High	High	False	Yes
Rain	Mid	High	False	Yes
Rain	Low	Low	False	Yes
Cloudy	Low	Low	True	Yes
Sunny	Low	Low	False	Yes
Rain	Mid	Low	False	Yes
Sunny	Mid	Low	True	Yes
Cloudy	Mid	High	True	Yes
Cloudy	High	Low	False	Yes
Sunny	High	High	False	No
Sunny	High	High	True	No
Rain	Low	Low	True	No
Sunny	Mid	High	False	No
Rain	Mid	High	True	No

Now, we calculate H_{new} for every feature:

$$H_{new,sky} = \frac{9}{14}(-\frac{2}{9}\log_2(\frac{2}{9}) - \frac{4}{9}\log_2(\frac{4}{9}) - \frac{3}{9}\log_2(\frac{3}{9})) + \frac{5}{14}(-\frac{3}{5}\log_2(\frac{3}{5}) - \frac{2}{5}\log_2(\frac{2}{5})) = 1.33066$$

$$H_{new,temp} = \frac{9}{14}(-\frac{2}{9}\log_2(\frac{2}{9}) - \frac{4}{9}\log_2(\frac{4}{9}) - \frac{3}{9}\log_2(\frac{3}{9})) + \frac{5}{14}(-\frac{2}{5}\log_2(\frac{2}{5}) - \frac{2}{5}\log_2(\frac{2}{5})) - \frac{1}{5}\log_2(\frac{1}{5})) = 1.82597$$

$$H_{new,hum} = \frac{9}{14}(-\frac{3}{9}\log_2(\frac{3}{9}) - \frac{6}{9}\log_2(\frac{6}{9})) + \frac{5}{14}(-\frac{4}{5}\log_2(\frac{4}{5}) - \frac{1}{5}\log_2(\frac{1}{5})) = 0.84816$$

$$H_{new,wind} = \frac{9}{14}(-\frac{3}{9}\log_2(\frac{3}{9}) - \frac{6}{9}\log_2(\frac{6}{9})) + \frac{5}{14}(-\frac{3}{5}\log_2(\frac{3}{5}) - \frac{2}{5}\log_2(\frac{2}{5})) = 0.93710$$

Next, we can calculate the information gain for each of the features:

$$I_{sky} = H_{old,sky} - H_{new,sky} = 1.57741 - 1.33066 = 0.2468$$

$$I_{temp} = H_{old,temp} - H_{new,temp} = 1.55666 - 1.82597 = -0.2693$$

$$I_{hum} = H_{old,hum} - H_{new,hum} = 1 - 0.84816 = 0.1518$$

$$I_{wind} = H_{old,wind} - H_{new,wind} = 0.98523 - 0.93710 = 0.04813$$

With all the information gains calculated, we can finally rank the features based on their relevance in determining whether the cyclist went on a ride or not:

Rank	Feature	Information gain
1	Sky condition	0.2468
2	Humidity	0.1518
3	Windy	0.04813
4	Temperature	-0.2693

3.2.1

Found in the Matlab script MyImpute.m is a function which takes in a matrix of data that contains missing and a vector stating whether each column of the data matrix is a Categorical or Continuous variable and outputs the same data matrix with imputations for the missing data base on the mean for continuous data, and the mode for categorical data.

For the input of the data matrix it must either be of type 'table' or in matrix form in which case matlab will automatically convert all entries to strings (see below for input example). For s the the vector which indicates the type of each column input should simply be a vector containing "Cont" for Continuous variables and "Cat" for Categorical variables. Anything other than this and imputation will not be carried out for the associated column.

Input example:

data = ["X","A",6;"Y","B",NaN;NaN,NaN,NaN,NaN;"X","C",8;"Y","D",12]

s = ["Cat","Cat","Cont"]

Note: input of type 'table' also accepted however missing values must be in as <missing>

In the case when there are two or more candidates for the mode the matlab mode function will automatically pick the candidate out of the possible modes which comes which comes first alphabetically. Ideally on each imputation a random choice between the modes (with each having a equal chance of being chosen) would be a better option however the built in matlab mode function does not accomodate and a complete new function for the mode would need to be written in order to do this.

3.2.2

a) Feature selection via the feature shuffling method consists of training a classifier against the original data set, permuting the values of each feature (column) of the data set and then calculating the accuracy after applying the classifier. The below is a Matlab feature shuffling function from featureselection.m. It takes the original data set X and y together with the classifier trained on the original data set. First it creates d copies of X where d is the number of features in X. Then it permutes the elements of the ith feature in the ith copy of X. Finally, the model is applied to the new data sets and the accuracy is determined for each new data set.

```
function acc = featureshuffling(X, y, mdl)
    [n, d] = size(X);
    acc = size(1,d);

    %create d copies of X
    perm = repmat(X, 1, 1, d);
    % shuffle the ith column of the ith copy of X
    for i = 1:d
        perm(:,i,i) = X(randperm(n),i);
    end
    % test the classifier on the copies of X
    % calculate the accuracy of the classifier
    % on the new data
    for j = 1:d
        [ypred, ~] = predict(mdl, perm(:,:,j));
        accuracy = mean(ypred==y);
        acc(j)=accuracy ;
    end
end
```

b) The importance of the features is determined from the difference between the accuracy of the original data set and the average accuracy of the permuted data set. Features which are heavily correlated with the diabetes column can be identified by the fact that after permuting their values, the corresponding accuracy changes drastically. From Figure 1 one can clearly note that glucose levels and BMI are the most important factors when determining whether one has or will develop diabetes. Of course, if one uses other classifiers one may end up with different values of importance and the feature rankings may change. Another important thing to note is that running this script gives slightly different results each time since the permutations are random. One way this could be improved is by running for all possible permutations and then taking an average although that would be quite computationally expensive.

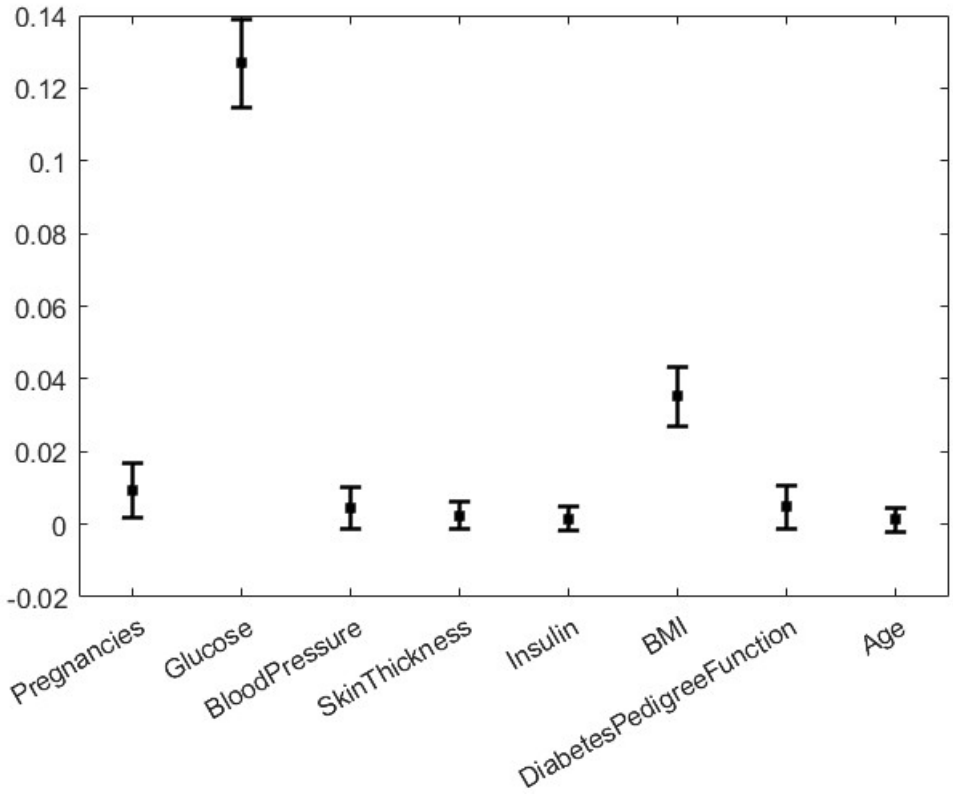

Figure 1: Plot of the importance of each feature when determining diabetes.