

Data pre-processing - 2

Dr Mohamed Bennasar

Outline

Data Pre-processing

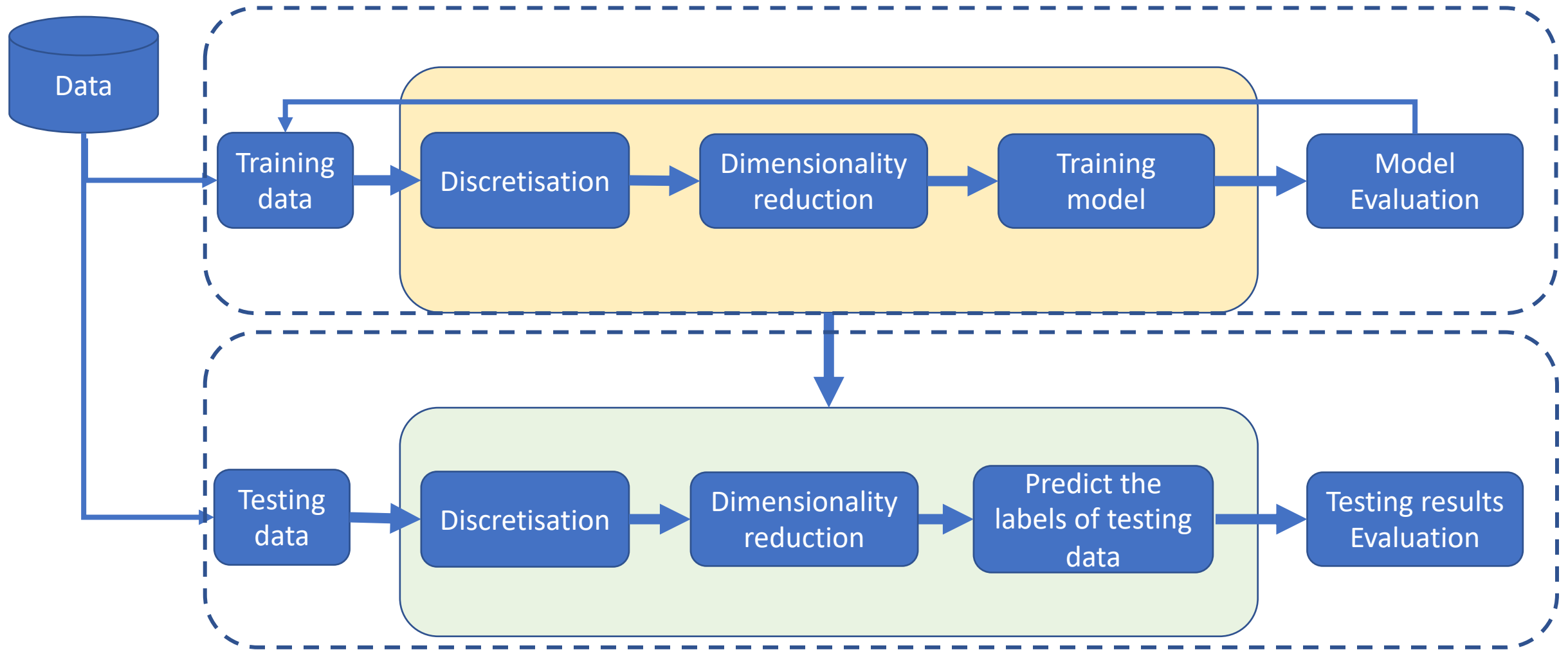
Dimensionality Reduction

Data transformation: Discretisation

Imbalanced Data

Discretisation

Supervised ML pipeline



Data transformation (Discretisation):

- Data transformation is the process of transforming data from one format to another that is more appropriate for machine learning.
- Discretisation is a process that transforms continuous numerical data into data can only take certain values without significant loss of information.
- Discretisation changes a numeric input variable to have a different data distribution

Why is discretisation needed ?

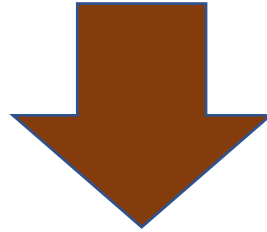
- Some machine learning algorithms may prefer or require discrete.
- It can speed up the learning by discretising the search space needed by the learning algorithm.
- It can also lead to a greater increase in the accuracy of the Naïve Bayes classifier
- It also performs a summary of the continuous data
- allows learning algorithms to benefit from the relationships between the attributes and to use continuous and discrete attributes together in a more reliable manner.

Simple example

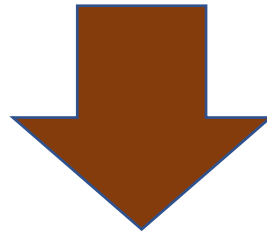
Suppose that we have one feature datasets which is giving the age of 25 people, with range of [18–82].

We would like to divide the range of the ages into three groups: young (Y), middle age (MA), and old (O), where $Y = [18-30]$, $MA = [31-50]$, and $O = [51-82]$.

18	25	20	50	55	40	35	65	19	38	37	41	42	26	24	22	33	39	23	70	80	20	73	69	34
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----



Y=[19-30], MA=[31-50], and O= [51-80]



Y	Y	Y	MA	O	MA	MA	O	Y	MA	MA	MA	MA	Y	Y	Y	MA	MA	Y	O	O	Y	O	O	MA
---	---	---	----	---	----	----	---	---	----	----	----	----	---	---	---	----	----	---	---	---	---	---	---	----

Discretisation techniques

These techniques search for the most appropriate boundaries of the discretisation intervals. These techniques can be classified into:

- **Dynamic variable discretizing** involves discretizing all the variables at once, or simultaneously. In dynamic variable discretization we have to keep track of, and deal with, any interdependencies (interactions) between the variables.
- **Unsupervised discretization algorithms** are the simplest algorithms to make use of, because the only parameter you would specify is the number of intervals to use; or else, how many values should be included in each interval.
- **Supervised discretization algorithms** don't specify the number of bins, and the discretization is run based on entropy and purity based calculations.

Information theory-based method

The ideal way to discretise some data is to ensure that each bin we use corresponds to one class label. Of course, this is rarely possible, but the closer we can get to this idea, the better.

One way to determine how close we are to the ideal is to find the mutual information between our discretisation bins and the class labels.

Information theory-based method

1. Calculating the entropy of the class label
2. suggesting a cut point/s to divide the feature into bins. Let us assume the cut point is 60.

		Class	
		Y	N
Feature	<=60	3	0
	>60	4	17

3. Calculating the mutual information between the class and the discretised feature
4. Then this process is repeated until we get the highest mutual information between the class and the discretised feature.

Feature (F)	Class label (C)
53	Y
56	Y
57	Y
63	N
66	N
67	N
67	N
67	N
68	N
69	N
70	N
70	Y
70	Y
70	Y
72	N
73	N
75	N
75	Y
76	N
76	N
78	N
79	N
80	N
81	N

Equal Width Discretisation (EWD)

This divides the range of the feature's values into a fixed number of bins of equal width.

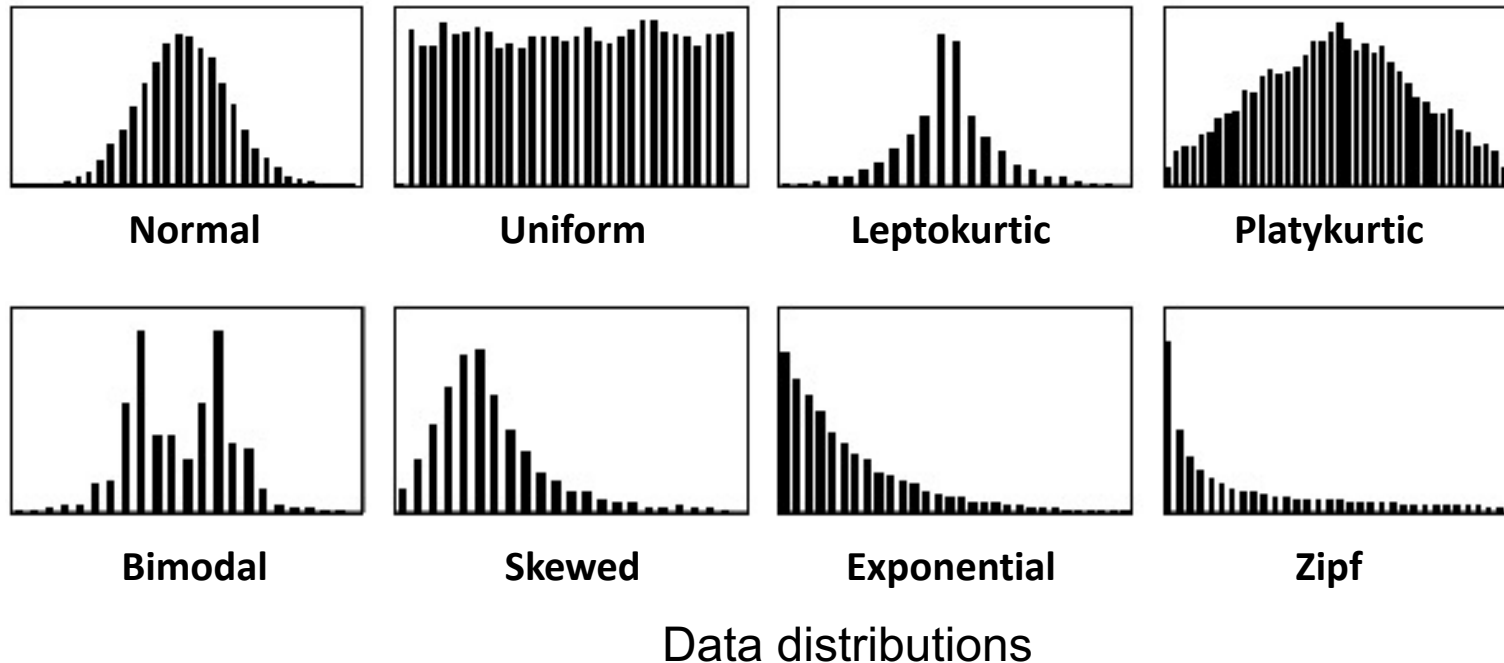
The user must specify the number of intervals N . If X and B are the lowest and highest values of a continuous feature, respectively, then the interval width w is

$$w = \frac{y - x}{N}$$

The cut-points will be : $X + w, X + 2w, \dots, X + (N - 1)w$

Issues with EWD

1. The method performs well with data that has uniform distributions.
2. It is vulnerable to skewed data and does not perform well if data has a normal distribution.
3. It is vulnerable to outliers



Equal Frequency Discretisation (EFD)

The continuous attribute range is divided into N intervals, each one containing the same number of instances.

This method solves the problem of imbalanced intervals.

Issues with EFD

It may separate similar attribute values into different intervals and group dissimilar attributes into the same interval.

Simple Example

Assume we have a single vector of data $X = [0, 4, 12, 16, 17, 18, 24, 28, 30]$.
Let us say that we want to use EWD and EFD to discretise this data into three bins.

Interval width of EWD will be $(30-0)/3 = 10$

EWD	
Interval 1: 0, 4	[0, 10)
Interval 2: 12, 16, 17, 18	[10, 20)
Interval 3: 24, 26, 30	[20, 30]
EFD	
Interval 1: 0, 4, 12	[0, 14)
Interval 2: 16, 17, 18	[14, 21)
Interval 3: 24, 28, 30	[21, 30]

Imbalanced data

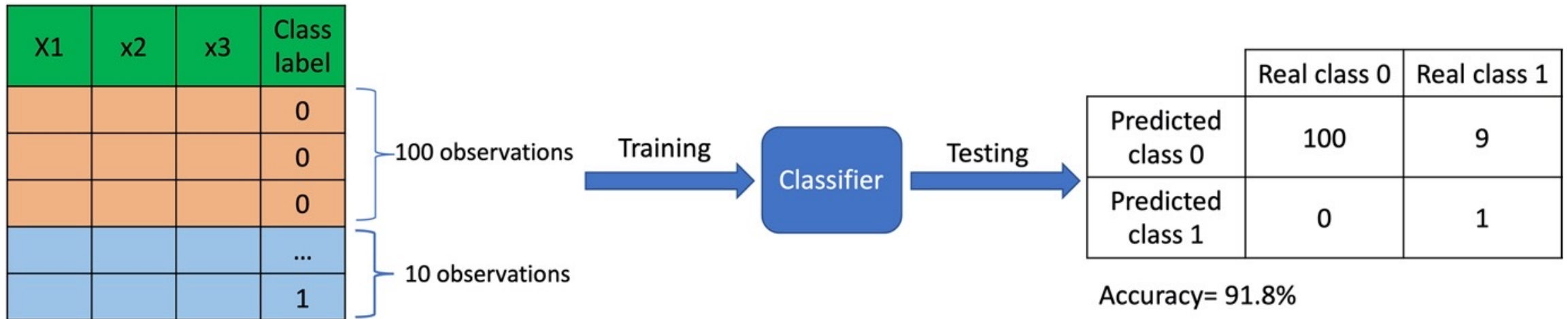
Imbalanced data

1. An imbalanced nature is a characteristic of many real-world data sets in many domains.
2. Imbalance appears when the classes in the data set are not evenly distributed.
3. The classes in the imbalanced data can be divided into majority classes, which have a large quantity of data, and minority classes, which have just a few instances of data.
4. The minority class is usually the most important in such cases.
5. Uneven distribution of classes can be very problematic for classification algorithms; causing biased performance since algorithms.

The accuracy paradox

Choosing the right metric to measure the performance (e.g. Classification accuracy is not the best choice).

let us assume we have imbalanced data set that consists of a skewed binary class label (100 of class 0, and 10 of class 1)



The accuracy paradox

- Classification accuracy as metric focuses on the true positives and the true negatives only.
- The total accuracy is 91.8%, which would appear as a good performance.
- The classifier failed to predict 90% of the observations of the class 0 and misclassified as class 1.
- Other metrics such as True Positive Rate (TPR), and True Negative Rate (TNR) can be used to evaluate the classification performance reliably.

The accuracy paradox

$$TPR = \frac{TP}{TP + FN} = \frac{100}{100 + 1} = 0.99$$

$$TNR = \frac{TN}{TN + FP} = \frac{1}{9 + 1} = 0.1$$

It is clear that negative rate metric is able capture the true performance of the classifier in predicting the minority class.

Solutions for the classification of imbalanced data

The imbalanced data issue can be dealt with at two levels:

1. Data-level solution

- In this approach, resampling techniques are employed to increase the number of fraction minority class observations.
- This approach is categorised into three groups, namely over-sampling , under-sampling, and data augmentation.

Synthetic Minority Over-sampling Technique (SMOTE)

This is an over-sampling method which works in feature space

The method performs the over-sampling through the following steps:

1. Identifying the minority class data points using the feature vector and defining the nearest neighbour.
2. Calculating the distance between a pair of data points of the minority class.
3. Multiplying the distance with a random number between 0 and 1.
4. Add new data points on the line segment between the original data points by adding the random number to the feature vector.

Algorithm-level solution

The classification algorithm is modified to handle the difference in the number of observations for each class during the training process.

This approach includes cost-sensitivity-based techniques and ensemble techniques:

Cost-sensitivity-based techniques : involves using hyperparameters to help the classifiers deal internally with imbalanced data by giving more weight to the minority class.

Ensemble techniques: using more than one classifier, each one is trained with sample of the majority class and the whole datapoints of the minority class