Data Preprocessing

Click on a question number to see how your answers were marked and, where available, full solutions.

Question Number	Score		
Question 1	1	/	1
Question 2	1	/	1
Question 3	0	/	0
Question 4	1	/	1
Total	**3**	**/**	**3 (100%)**

The pass rate for the questions in the tutorial is 50%, if you score less than this you might want to revisit the questions you had difficulty with and read some of the resources pertaining to that topic.

Thank you for using this tool, in order to improve the system please complete the questionnare linked below:

https://forms.ncl.ac.uk/view.php?id=6719176 (https://forms.ncl.ac.uk/view.php?id=6719176)

Performance Summary

Exam Name:	Data Preprocessing
Session ID:	13356001884
Exam Start:	Thu Dec 10 2020 11:46:38
Exam Stop:	Thu Dec 10 2020 11:46:55
Time Spent:	0:00:16

Question 1

What do we mean when we talk about data?

Data consists of recorded information, observations or values.

What do we mean when we talk about variables?

A variable is any entity that can have a number of values. There are three types of variable:

Discrete: countable or fixed set of values, e.g. number of students present, number of buttons in a jar.

Continuous: uncountable values, a continuous variable can take on a range of values between points of measurement, examples include speed and distance.

Categorical: variables whose values are labels rather than numerical, e.g. a "bird" variable with the values "pigeon", "sparrow" and "seagull".

Frequency (f):

The number of times when a particular value occurs.

Frequency Distribution:

The frequency of values in a sample, the frequency distribution is usually displayed in a list or a graph such as a histogram.

The figure below displays the data and the table containing the variables (x) and the frequency distribution (labelled f).

18	12	14	17	19	12
21	18	16	17	12	14
13	22	12	18	14	13

Variable (x)	Frequency (f)
12	4
13	2
14	3
16	1
17	2
18	3
19	1
21	1
22	1

Would the following be a discrete or continuous variable:

The length of time it takes a truck driver to drive from Sunderland to London

Discrete [✓ Continuous]

Expected answer:

Discrete [✓ Continuous]

✓

✓ You chose a correct answer. You were awarded **1** mark.

You scored **1** mark for this part.

Score: 1/1 ✓

Question 2

Tensors

Tensors are fundamental to the field of machine learning as most ML systems use tensors as their data structure.

A tensor is a data container.

Tensors are a generalised version of matrices as they can be an arbitrary number of dimensions (referred to as *axis* in relation to tensors)

A scalar is a 0D tensor

A vector is a 1D tensor

A matrix is a 2D tensor

A tensor has 3 key attributes:

- Number of axes: for example, a matrix has two axes. Within Python libraries such as Numpy the number of axes is called `ndim`.
- Shape: this details the number of dimensions of the tensor along each axis. In Numpy a vector would have a shape with a single element for example `(3,)`. A scalar would have an empty shape, `()`. The matrix below would have the shape `(3,4)`

$$\begin{bmatrix} 2 & 51 & 19 & 4 \\ 27 & 14 & 5 & 9 \\ 30 & 42 & 21 & 7 \end{bmatrix}$$

- Data type: This details the kind of data contained in the tensor. This is usually called `dtype` in Python libraries. Example tensor data types include `float32`, `float64` and `uint8`.

Examples of data tensors

The data you're likely to work with in your future machine learning projects can be categorised into one of the following:

- *Vector data* - 2D tensors of the shape `(samples, features)`
- *Timeseries data or sequence data* - 3D tensors of shape `(samples, timesteps, features)`
- *Images*- 4D tensors of shape `(samples, height, width, channels)` or `(samples, channels, height, width)`
- *Video* - 5D tensors of shape `(samples, frames, height, width, channels)` or `(samples, frames, channels, height, width)`

(Data tensors examples from: F.Chollet (2018) Deep Learning with Python. New York: Manning Publications)

How many dimensions would a timeseries data tensor contain?

2D [✓ 3D] 4D

Expected answer:

2D [✓ 3D] 4D

✓

✓ You chose a correct answer. You were awarded **1** mark.

You scored **1** mark for this part.

Score: 1/1 ✓

Question 3

The data to be used within your machine learning algorithm is often not in a suitable condition to be plugged straight into your model, it is often necessary to undertake some processing to tranform the data into a compatible format.

There are usually three stages to prepare your data for machine learning:

1. Selecting the data
2. Preprocessing the data
3. Feature engineering

Selecting the Data

The quality of the data used within your machine learning algorithm is a key determinant of how well your algorithm will learn. When choosing the subset of data from the larger dataset it is essential to:

- Define the problem you are working on?
- Determine the data that is essential to solving this problem?
- Think of your strategy to solve the problem?
- Choose which machine learing techniques you are going to use?

These stages are often repeated/iterated over.

Preprocessing the Data

- Formatting - the data needs to be in a suitable format for you to work with.
- Cleaning - missing values in our data can be identified as either `NULL` or `NaN` (not a number). Once we have found the missing values we can either drop the row or column which contains the missing values or input a value using an interpolation technique. The most common of these techniques is mean imputation, where the missing value is replaced with the value of the mean of the feature column. This technique can be implemented using scikit-learn `Imputer` class.
- Sampling - for the sake of efficiency when trialling and prototyping your solutions, it is beneficial to take a smaller representative sample of the overall dataset to initially work with.

The preprocessing of data and techniques such as cleaning and sampling are ongoing open areas of research.

What is a feature?

In machine learning a feature is a measurable, descriptive attribute. For example it could be a column heading in your dataset such as height, age or gender.

Feature Engineering

- Representation transformation - is the the process of mapping data to useful features, this process will be dealt with in further detail in the ***Data Representation*** tutorial.
- Feature tuning- the majority of machine learning algorithms respond much better if the features are on a similar scale. At this stage, perform normalization or standardization if necessary. *Normalization* changes the values of numeric data in your dataset to a common scale, typically [0,1]. *Standardization* typically means rescaling the data to have a mean of 0 and a standard deviation of 1.
- Feature extraction - reducing the number of features to create more insightful data representations, algorithms such as PCA (Principle Component Analysis) can be utilised for this purpose.
- Feature decomposition - Within your dataset you may have features which serve a complex concept, feature decomposition is the process of splitting this into constituent parts.
- Aggregation - is the opposite of feature decomposition in that you may have features which can be combined into a single feature.
- Instances selection - creating the training, validation and test sets from the dataset. One of the easiest ways to implement this is using scikit learn `sklearn.model_selection.train_test_split`

Resources:

https://machinelearningmastery.com/how-to-prepare-data-for-machine-learning (https://machinelearningmastery.com/how-to-prepare-data-for-machine-learning/)/ (https://machinelearningmastery.com/how-to-prepare-data-for-machine-learning/)

https://medium.com/@urvashilluniya/why-data-normalization-is-necessary-for-machine-learning-models-681b65a05029 (https://medium.com/@urvashilluniya/why-data-normalization-is-necessary-for-machine-learning-models-681b65a05029)

https://towardsdatascience.com/normalization-vs-standardization-quantitative-analysis-a91e8a79cebf (https://towardsdatascience.com/normalization-vs-standardization-quantitative-analysis-a91e8a79cebf)

Question 4

Data representation is the process of mapping data to useful features.

There are a number of differing mappings you may want such as:

Numerical representation - e.g. represent images as arrays of pixel data

Categorical to numerical - e.g. One Hot Encoding

One Hot Encoding **-** this is used to transform *categorical features into numerical variables*. This technique is commonly used when the learning algorithm is particularly for numerical functions.

For example - we have the feature `colour` which holds the values `orange = 0, green = 1, purple = 0`

If we use this data within a learning algorithm an assumption will be made that green is larger than orange and purple, allowing the algorithm to assume a natural ordering can lead to poor results.

We can use one hot encoding to create a feature for each unique value for the `colour` feature column. The three new features would be `orange, green and purple`.

A particular colour of a sample can then be indicated with a binary value. For example an `orange` sample can be encoded as `orange =1, green =0, purple =0`

One hot encoding can be performed in Python using the `OneHotEncoder` in the `scikit-learn.preprocessing` module.

Numerical to categorical - e.g. `grade{42, 60, 45, 77, 84}` transformed into `{'pass', 'merit', 'pass', 'distinction', 'distinction'}`

If we had: `age{1,5, 7, 18, 54, 25}` and transformed it into `{'infant', 'child', 'child', 'adult', 'adult','adult'}`

What type of mapping have we undertaken?

○ Categorical to numerical ○ Numerical representation

[✓ Numerical to categorical]

✓

Expected answer:

Categorical to numerical Numerical representation [✓ Numerical to categorical]

✓ You chose a correct answer. You were awarded **1** mark.

You scored **1** mark for this part.

Score: 1/1 ✓

Created using Numbas (https://www.numbas.org.uk), developed by Newcastle University (http://www.newcastle.ac.uk).