School of Computing and Information Systems
The University of Melbourne
COMP90049 Introduction to Machine Learning (Semester 2, 2022)
Sample Solution Week 2

Considering the following problems:

(i) Skin cancer screening test

(ii) Building a system that guesses what the weather (temperature, precipitation, etc.) will be like tomorrow

(iii) Predicting products that a customer would be interested in buying, based on other purchases that customer has previously made

1. Identify the "concept" we might attempt to "learn" for each problem (Task Identification)

   What we trying to learn in usually the parameter (or concept) that we are trying to predict or understand (using a Machin Learning technique). It is the final output of the system which can be a label (such as sunny, rainy, cloudy) or a quantity (like the possible temperature) or a cluster (like spam / not-spam) or something else (e.g., an association rule). In a *supervised learning* problem (such as classification or regression) this concept usually referred to as a *label* or the *response variable.*

   As for our sample problems:

   (i). Since it is a screening test, we are trying to answer whether a patient has a cancer or not. It is a binary decision (True or False) which it is a very common in Machine Learning.

   (ii). Various weather features of the particular day (that we are trying to predict) can be considered as the output of the system. The prediction can be a quantity like the temperature or amount of rain or the UV index or any other weather feature.

   (iii). There are two approaches to this problem:

      a. We want to exhaustively label every product for every customer as either "interested" or "not interested". For example, we know that for the past 6 months the customer purchased 'milk' (in average) every 7 days. So, we can predict if the customer would be "interested" in purchasing 'milk' next time (s)he enters the store.

      b. We want to predict if our customer would be interested in a single product (or set of products) that (s)he has not purchased before. In this situation we can find the group of customers that have similar purchase habits/taste and based on their purchase history predict the behaviour of our particular customer.

2. For each problem-task, identify what the instances and attributes might consist of (choosing the data representative)

   An instance is a single exemplar from the data, consisting of a bundle of (possibly unknown) attribute values (feature values) [and in the case of supervised ML a class value].

   An attribute is a single measurement of some aspect of an instance, for example, the frequency of some event related to this instance, or the label of some meaningful category.

   Attributes are usually classified as either nominal (labels with no ordering), ordinal (labels with an ordering), or continuous (numbers, even if they perhaps aren't continuous in the mathematical sense).

(i). In this case each patient is an instance. The attributes can be results of the blood test, images from the skin, reports, observed syndromes and so on.

(ii). It seems fairly clear that each instance will be a day; depending on how we construe the problem, various properties could be attributes — the most logical is probably the corresponding data (temperature, precipitation, humidity, wind speed, etc.) from the previous day(s).

(iii). For scenario (a) of last question, customer-product pairing can be the possible instance. For scenario (b) each customer would be an instance. In either way the attributes can be the customer's name, age, address, gender, shopping log, credit card information, loyalty card information and more.

3. For each problem-task, conjecture whether a typical strategy is likely to use supervised or unsupervised Machine Learning (picking a suitable model)

Generally speaking, supervised techniques in machine learning start from exemplars (instances) — labelled with classes — in a set of training data and use these to classify unknown instances in a set of test data.

Unsupervised methods are not based on a set of labelled training data. Unsupervised methods often broken down into 'weakly unsupervised methods' (where the class set is known, but the system does not have access to labelled training data), and 'strongly unsupervised methods' (where even the class set is unknown, and we don't even know how many classes we have).

(i). Assuming that we have trained our model based on the historical data from previous patients, it would be a (binary) **Classification** problem.

(ii). For this problem, assuming that we can access historical data for the particular location, (supervised) **regression** seems like the most plausible ML strategy. So, we find the pattern using the attributes value from previous days, months and years and predict our weather feature (e.g. temperature). This case could potentially also be **classification** – instead of predicting the temperature, wind speed, etc. we can just give one label like on a weather app ("Sunny," "Rainy," etc.).

(iii). For our two different approaches:

a. In this scenario, we have a **classification** problem, where we might try to predict "interested" "not-interested" labels based on some properties of the product and customer. Classification is a supervised learning method.

b. It can be a (unsupervised) **clustering** method, where we find groups (clusters) of customer with same features; or an **association rule mining** method that we identify an association between customer(s) and some attribute(s) in the products. (e.g., if the product is from 'Nestle' there is x% probability that customers age groups of A and B would purchase it.)

4. [OPTIONAL] For each problem-task, consider how easy or difficult it would be to make a model that generalizes to new cases. For example, could you predict the weather in any city in the world, or just in one specific city?

(i). Generalization is a big concern for machine learning in the medical domain, because real world training data often have biases, and these biases can affect performance in various ways. For example, we know skin cancer risk increases with age, so these variables will probably be correlated in your training set. On the one hand, this is good – the model should correctly learn that age predicts skin cancer. But it can also be bad if the model becomes too dependent on that predictor (e.g., if it decides to label every image of older-looking skin as "cancer"). And if there were very few instances of young people with cancer in the training set, the resulting model might not work well on younger patients.

(ii). The weather model might work better in some cities than others. It would probably generalize better if it included geographic information in addition to the previous days' weather (e.g., longitude, altitude, distance from ocean, distance to mountains) because then it could learn how these features interact with the weather patterns.

(iii). A customer model trained in one country might not generalise to other countries. If it mostly learns everyday shopping patterns, it probably won't give good predictions for outlier situations like holiday purchasing.

5. [OPTIONAL] What kinds of assumptions might a machine learning model make when tackling these problems?

Every model makes assumptions about the world and how the concepts we want to learn relate to the attributes of the data.

The first assumption we make is that the concept is actually related to the attributes! This assumption is so obvious that we rarely discuss it – usually we only include attributes that we think are likely to predict the concept. For example, you would probably not use "patient's favourite song" as an attribute for skin cancer detection. However, this attribute might actually be a good predictor, because your favourite song can be a good predictor of your age, and age is a risk factor for skin cancer. You could probably come up with other "weird" predictors for each of the example models.

Secondly, each model makes assumptions about the ways the attributes can relate to the concepts. For example, does it make more sense for the models to treat all attributes as independent predictors, or would it be better to use a model that allows the predictors to interact? In most of these cases we would expect the attributes to interact in complex ways but allowing interactions could lead to an overly complex model in the cases where there are many attributes to start with (for example, in the customer purchasing model). For the problems with numeric attributes, we would generally expect linear (or monotonic, e.g., strictly increasing or decreasing) relationships between the attributes and concepts. This is often a good simplifying assumption for machine learning, but it limits what a model can learn. For example, the relationship between a product and price might be U-shaped – very cheap and very expensive products might be less popular than products priced somewhere in the middle.

6. What is **discretisation**, and where might it be used? Discretise attribute C of the following dataset according to the given methods (breaking ties where necessary).

We have a (continuous) numeric attribute, but we wish to have a nominal (or ordinal) attribute. Some datasets inherently have groupings of values, where treating them as an equivalent might make it easier to identify underlying patterns.

ID	A (°C)	B (mm)	C (hPa)	CLASS
1	22.5	4.6	1021.2	AUT
2	16.7	21.6	1027.0	AUT
3	29.6	0.0	1012.5	SUM
4	33.0	0.0	1010.4	SUM
5	13.2	16.4	1019.5	SPR
6	14.9	8.6	1016.4	SPR
7	18.3	7.8	995.4	WIN
8	16.0	5.6	1012.8	WIN

(i) Equal width

Equal width divides the range of possible values seen in the training set into equally– sized sub-divisions, regardless of the number of instances (sometimes 0!) in each division.

– For attribute C above, the largest value is 1027.0 and the smallest value is 995.4; the difference is 31.6:

- If we wanted two "buckets", each bucket would be 15.8 wide, so that instances between 995.4 and 1011.2 take one value (4 and 7), and instances between 1011.2 and 1027.0 take another (1, 2, 3, 5, 6, and 8).
- If we wanted three "buckets", each bucket would be about 10.5 wide; so that instances between 995.4 and 1005.9 take one value (just 7), instances between 1005.9 and 1016.4 take another value (3, 4, 6, and 8), and instances between 1016.4 and 1027.0 take yet another value (1, 2, and 5).

(ii) Equal Frequency

Equal frequency divides the range of possible values seen in the training set, such that (roughly) the same number of instances appear in each bucket.

- For attribute C above, if we sort the instances in ascending order, we find 7, 4, 3, 8, 6, 5, 1, 2.
- If we wanted two "buckets", each bucket would have four instances; so that instances 7, 4, 3, and 8 would take one value, and the rest would take the other value.
- Sometimes, we also need to explicitly define the ranges, in case we obtain new instances later that we need to transform. There is some question about the intermediate values (between 1012.8 and 1016.4, in this case); typically, we place the dividing point at the median.

(iii) k-Means

k-means is actually a "clustering" approach, but it can work well in this context. If we want k buckets, we randomly choose *k* points to act as seeds. We then have an iterative approach where we: assign each instance to the bucket of the closest seed; update the "centroid" of the bucket with the mean of the values.

- For attribute C above, let's say we begin with two random seeds: instance 3 (1012.5, bucket A) and instance 4 (1010.4, bucket B).

A (1012.5)	B (1010.4)

- Instance 1 (1021.2) is closer to A than B; 2 (1027.0) is closer to A; 3 is closer to A; 4 is closer to B; 5 (1019.5) is closer to A; 6 (1016.4) is closer to A; 7 (995.4) is closer to B; 8 (1012.8) is closer to A.

A (1012.5)	B (1010.4)
1	4
2	7
3	
5	
6	
8	

- We take the average of the values of instances 1, 2, 3, 5, 6, and 8 (1018.2) to be representative of cluster A, and the average of instances 4 and 7 (1002.9) to be representative of cluster B.
- Now, we iterate: 1 is still closer to A; 2 is still closer to A; 3 is still closer to A; 4 is still closer to B; 5 is still closer to A; 6 is still closer to A; 7 is still closer to B; 8 is still closer to A.

A (1018.2)	B (1002.9)
1	4
2	7
3	
5	
6	
8	

– Since this is the same assignment of values to clusters, we stop: instances 4 and 7 will have one value, and the other instances will have another value.