University of London

Assessment Coversheet

Complete this coversheet and read the instructions below carefully.

Candidate Number: KJ1932
Refer to your Admission Notice

Degree Title: BSc in Computer Science
e.g. LLB

Course/Module Title: Data Science
As it appears on the question paper

Course/Module Code: CM3005
This is in the top right corner of the question paper. If there is more than one code, use the first code.

Enter the numbers, and sub-sections, of the questions in the order in which you have attempted them:

Question 2 a b c d
Question 3 a b c d

Date: March 16 2022

Instructions to Candidates

1. Complete this coversheet and begin typing your answers on the page below, or, submit the coversheet with your handwritten answers (where handwritten answers are permitted or required as part of your online timed assessment).
2. Clearly state the question number, and any sub-sections, at the beginning of each answer and also note them in the space provided above.
3. For typed answers, use a plain font such as Arial or Calibri and font size 11 or larger.
4. Where permission has been given in advance, handwritten answers (including diagrams or mathematical formulae) must be done on light coloured paper using blue or black ink.
5. Reference your diagrams in your typed answers. Label diagrams clearly.

The Examiners will attach great importance to legibility, accuracy and clarity of expression.

Begin your answers on this page

Question 2

(a) The precision and recall values need to be calculated for each class by adding the values of the other two classes together as the negative class.

The formulas used are:

Precision = True Positives / (True Positives + False Positives)
Recall = True Positives / (True Positives + False Negatives)

I will round values to two decimal places.

For the class "**Dog**":

- **Precision** = 20 / (20 + 3 + 1) = **0.83**
- **Recall** = 20 / (20 + 6 + 4) = **0.67**

For the class "**Cat**":

- **Precision** = 6 / (6 + 16 + 2) = **0.25**
- **Recall** = 6 / (6 + 20 + 4) = **0.2**

For the class "**Rabbit**":

- **Precision** = 4 / (4 + 1 + 7) = **0.33**
- **Recall** = 4 / (4 + 20 + 6) = **0.13**

The average precision and recall are calculated as weighted means using the true values of each class.

- **Average Precision** = [(20 * 0.83) + (6 * 0.25) + (4 * 0.33)] / (20 + 6 + 4) = **0.65**
- **Average Recall** = [(20 * 0.67) + (6 * 0.2) + (4 * 0.13)] / (20 + 6 + 4) = **0.50**

The F1-Score for each class is calculated as follows.

F1 = 2 * ((precision * recall) / (precision + recall))

- **F1 Dog** = 2 * ((0.83 * 0.67) / (0.83 + 0.67)) = **0.74**
- **F1 Cat** = 2 * ((0.25 * 0.2) / (0.25 + 0.2)) = **0.22**
- **F1 Rabbit** = 2 * ((0.33* 0.13) / (0.33 + 0.13)) = **0.19**

(b) Because the cost of suggesting a high risk surgery to a patient is much higher than a "false alarm", accuracy is not a good metric in this situation.

I would suggest using recall and precision. Recall would help us understand how many of the actual rare conditions we detected, and precision would help us understand how often our prediction of the rare condition was actually true. In this scenario, we would want to maximize precision first, and recall can serve as a guardrail that tells us how well our model performs over time.

(c) We could fill any missing values with a **mask**, e.g. a boolean flag, or a sentinel value, e.g. -999.

Masks would require adding a new column, and sentinel values require us to deal with the fact that computations might be affected.

We could also decide to drop any null value data instead of filling.

Other filling strategies include:

- filling with a previous value
- filling with the most frequent value
- filling with a mean value
- filling with an imputed value, by interpolating between values

(d) We can assume that fitness scores correlate with age. Hence, a mean value of fitness will not produce reliable data for all ages, but only for the mean age in the data set. For example, a mean value of 50 (out of 100) as a fitness score will dramatically overestimate the fitness of an 80-year old person and underestimate it for younger people.

What might be useful is imputing the fitness scores linearly by first fitting a function that maps age with fitness. Then, this function can be used to calculate the expected fitness score for a given age.

Question 3

(a) The prior probability is the probability of a hypothesis being true before any evidence was observed, denoted as P(H). The likelihood is the probability of seeing some evidence given that the hypothesis H is true, denoted as P(E|H).

In other words, the prior probability expresses how likely the hypothesis is to be true generally, while likelihood attaches the probability of the hypothesis being true to the presence of some data or evidence.

(b) We are looking for the probability that the pill was drawn from Bag 1 given the evidence that the pill was red, or P(Bag1|Red).

We know the probabilities of drawing red and blue pills given the bags:

P(Red | Bag1) = 4/10 = 2/5

P(Blue | Bag1) = 3/5

P(Red | Bag2) = 12/20 = 3/5

P(Blue | Bag2) = 8/20 = 2/5

The probabilities of drawing a red or blue pill overall are:

P(Red) = 16/30 = 8/15

P(Blue) = 7/15

Given Bayes Theorem, we can define the probability of drawing a pill from bag 1 as the prior, P(H).

P(H) = P(Bag1) = 10/30 = 1/3

We also define the likelihood as the probability of the pill being red given that we draw from bag 1, P(E|H), stated above as 2/5. The marginal likelihood P(E) is given by P(Red) above as 8/15.

Now we can apply Bayes theorem:

P(H|E) = (P(H) * P(E|H)) / P(E)

P(Bag1|Red) = (1/3 * 2/5) / (8/15) = 0.25 = 1/4

Therefore, the probability that a red pill drawn was drawn from Bag 1 is 25%, or 1 in 4.

(c)

We first define our null hypothesis that we want to disprove. In this case, the null hypothesis is that our coin is fair. Any deviations should be given by chance alone.

H0 = The coin is fair

P(H0) = 0.5

Let's set an expectation that we want 95% confidence in our final test. This means we are seeking a p-value of 0.05 or less.

Now we flip the coin a set number of times to obtain observations and record them. We can now check the deviation from the observations to our expected outcome. For example, if a coin was heads in 24 out of 50 flips, we have a deviation of 1 from the expected 25 heads.

We can look up this deviation in a Z-table to find the corresponding p-value. If it is less than 0.05, we can reject the null hypothesis and conclude that the coin is biased. If not, we can conclude that this is within the expected deviations and conclude that the coin is fair.

(d)

We want to find an optimal division of customers into clusters. To find optimal clustering, we want to find a cluster centre which is the arithmetic mean of all members, and each point is closer to its own cluster centre than to any others.

We can use K-Means clustering as an algorithm, which searches for a predetermined number of clusters within an unlabelled dataset.

We can apply the K-Means algorithm to both a 4-group and 5-group dataset and visualize the results, which can give us an indication of which clustering more accurately clusters our dataset into meaningful clusters with few outliers.

If we want to find an optimal division without knowing the number of groupings or clusters, we can use expectation maximisation. This assigns random cluster centres, assigns points to the nearest centre, then set the cluster centres to the mean. This is repeated to get a better estimate each time.