

ISC Class 12 Artificial Intelligence Specimen Paper 2

Written by the Future Topper subject team on the CISCE examination pattern. Not an official CISCE paper.
Found a mistake? Write to contactfuturetopper@gmail.com

Maximum Marks: 70 | Time Allowed: Three hours | Reading Time: 15 minutes

Paper 1 (Theory) carries 70 marks. Part I is compulsory. In Part II attempt one question from Section A, two from Section B and two from Section C.

General Instructions

Write your answers on the separate answer paper provided. The first 15 minutes are for reading the paper, and you may not write during that time. Part I is compulsory and carries 20 marks. Part II carries 50 marks and is divided into three Sections. From Section A answer any one of the two questions, from Section B answer any two of the three questions, and from Section C answer any two of the three questions. Marks for each question, or part of a question, are shown in brackets. Where a question asks for code, write it in Python; you may use the NumPy, Pandas, scikit-learn, Keras (TensorFlow), Matplotlib and OpenCV libraries named in the syllabus. Show all working for every calculation. The numbers have been chosen so that the arithmetic can be done by hand, and a value may be given correct to three decimal places unless the question says otherwise. No graphs, network diagrams, images, data tables or confusion matrices are printed in this paper; all data are given in words. Where a question asks for a diagram you are expected to draw it yourself in pencil and label it. This paper carries 70 of the 100 marks for the subject, the remaining 30 being for Paper 2 (Practical), which is made up of the practical examination and the practical file.

PART I (20 Marks)

Answer all questions. Question 1 carries 10 marks and Question 2 carries 10 marks.

Question 1 [10]

Each part carries 1 mark. For the parts that offer four choices, write only the option you choose. For the rest, answer in a single sentence or give the single word, name or value asked for.

- (i) Reducing the words "playing", "played" and "plays" to the common root "play" before words are counted is called:
(a) tokenisation (b) padding (c) stemming (d) embedding
- (ii) The correlation coefficient between daily hours of screen time and hours of sleep for a group of teenagers is -0.92. Interpret this value.
- (iii) For a normal distribution, the percentage of values that lie within one standard deviation of the mean is about:
(a) 50 per cent (b) 68 per cent (c) 95 per cent (d) 99.7 per cent
- (iv) During backward propagation, a neural network computes:
(a) the loss gradient for each weight (b) each neuron's output for a new input
(c) the number of hidden layers needed (d) the mean of each input feature
- (v) State what is meant by the stride of a convolution filter.
- (vi) After scikit-learn's StandardScaler has been fitted to the training data and applied to it, each feature has:
(a) minimum 0 and maximum 1 (b) mean 1 and standard deviation 0
(c) median 0 and range 1 (d) mean 0 and standard deviation 1
- (vii) Name the Keras method that is called to train a compiled model on the training data.
- (viii) A bank trains a model on past card transactions, each of which investigators have labelled genuine or fraudulent. This is an example of:

- (a) unsupervised clustering (b) supervised classification
 - (c) reinforcement learning (d) linear regression
- (ix) Giving each photograph of a road sign one label, such as stop, speed limit or no entry, is an example of:
- (a) object detection (b) object tracking
 - (c) image classification (d) facial recognition
- (x) Name the OpenCV function that opens a webcam or a video file so that its frames can be read one at a time.

Question 2 [10]

Each part carries 2 marks.

- (i) A plant disease app took the following times, in milliseconds, to classify eight leaf photographs: 120, 95, 130, 95, 110, 400, 105 and 95. Find the mean, the median and the mode, and state which of them best describes a typical response time, with a reason. [2]
- (ii) A grey scale image of 28 by 28 pixels is passed through a convolution layer of 8 filters, each 5 by 5, with stride 1 and no padding, followed by a 2 by 2 max pooling layer with stride 2. State the size of the output after each layer. [2]
- (iii) A column "state" holds the values Kerala, Assam and Punjab. Explain why one-hot encoding is preferred to label encoding (0, 1, 2) before this column is fed to a neural network. [2]
- (iv) A wind farm in Tamil Nadu plans to replace servicing every six months with predictive maintenance of its turbines. Give two advantages of predictive maintenance over fixed-schedule servicing. [2]
- (v) Give one reason why colour detection in OpenCV is usually done in the HSV colour space rather than in BGR. [2]

PART II

This Part carries 50 marks. Answer one question from Section A, two questions from Section B and two questions from Section C.

SECTION A (10 Marks)

Answer any one question from this Section. Each question carries 10 marks. This Section covers applications of AI: language processing, statistics and probability.

Question 3 [10]

- (a) A spam filter must turn each message into numbers before a model can learn from it. Give the output of the following program, and state why the word "a" does not appear in the vocabulary. [4]
- ```
from sklearn.feature_extraction.text import CountVectorizer
msgs = ["Win a free recharge now",
 "Call me now",
 "Free free entry, win now"]
cv = CountVectorizer()
X = cv.fit_transform(msgs)
print(X.shape)
print(cv.get_feature_names_out().tolist())
print(X.toarray()[2].tolist())
```
- (b) A news aggregator app shows stories from hundreds of Indian newspapers. Explain named entity recognition by giving two ways in which it helps the app, and describe one difficulty a recogniser faces, with an example. [3]
- (c) The message vectors from part (a), each labelled spam or not spam, are used to build a filter. Describe how the classifier is trained and tested, and explain why the filter must be retrained from time

to time. [3]

#### Question 4 [10]

- (a) The marks of candidates across a state in an AI aptitude test are normally distributed with mean 60 and standard deviation 8. Using the empirical rule, find the percentage of candidates who score between 52 and 76, and the percentage who score above 76. [3]
- (b) The national mean score in a reading test is 62 with a standard deviation of 8. A school in Lucknow introduces an AI reading app, and a random sample of 64 of its students has a mean score of 64.2. Test, at the 5 per cent level of significance, whether the mean of students using the app differs from the national mean. State the hypotheses, compute the test statistic  $z = (\text{sample mean} - \text{population mean}) / (\text{standard deviation} / \text{square root of } n)$ , take the critical value as 1.96, and state your conclusion. Name the error you might be making. [4]
- (c) A self-driving shuttle on a university campus leaves a stop every 20 minutes. A student reaches the stop at a random moment, so the waiting time is uniformly distributed between 0 and 20 minutes. Find the probability that the student waits more than 15 minutes, the probability that the wait is less than 8 minutes, and the mean waiting time. [3]

### SECTION B (20 Marks)

Answer any two questions from this Section. Each question carries 10 marks. This Section covers neural networks, machine learning and deep learning, and building and evaluating them in Python.

#### Question 5 [10]

- (a) A decision tree learnt from past sales predicts flat prices in a city. Its root asks whether the area is more than 1,000 square feet. If it is not, the tree asks whether the flat is within 2 km of a metro station: if yes it predicts 48 lakh rupees, and if no it predicts 35 lakh rupees. If the area is more than 1,000 square feet, the tree asks whether there are more than 2 bedrooms: if yes it predicts 95 lakh rupees, and if no it predicts 72 lakh rupees. Predict the price of flat P (850 square feet, 1.5 km from a metro station, 2 bedrooms) and flat Q (1,350 square feet, 6 km from a metro station, 3 bedrooms). Explain how this tree differs from a set of rules written by a programmer. [3]
- (b) A convolutional neural network is to classify colour photographs of 64 by 64 pixels as cat, dog or bird. Its first layer has 16 filters of size 3 by 3. Find the number of trainable parameters in this layer, and compare it with the number for a Dense layer of 16 neurons connected to every value of the flattened image. Use the comparison to explain why convolution suits images. [4]
- (c) A multi-layer perceptron uses a non-linear activation such as ReLU in its hidden layers. Explain what would happen if every layer used no activation at all, and why that matters for a task such as recognising handwritten digits. [3]

#### Question 6 [10]

- (a) Give the output of the following program, which carries out a forward pass of a small network, and state which line applies the activation function and what that function is. [4]

```
import numpy as np
x = np.array([1.0, 2.0])
W1 = np.array([[0.5, -1.0], [0.25, 0.5]])
b1 = np.array([0.0, -0.5])
z = x @ W1 + b1
h = np.maximum(0, z)
W2 = np.array([1.0, -2.0])
b2 = 0.5
y = h @ W2 + b2
print(z)
```

```
print(h)
print(y)
```

- (b) A model with one weight  $w$  predicts  $\hat{y} = wx$ , the weight multiplied by the input  $x$ . For a training example with  $x = 2$  and  $y = 5$ , the loss is  $L = (\hat{y} - y)^2$ , whose gradient is  $dL/dw = 2x(\hat{y} - y)$ . The weight is now 1.5 and the learning rate is 0.05. Carry out one step of gradient descent and show that the loss falls. [3]
- (c) Explain what is likely to happen during training if the learning rate is set far too high, and if it is set far too low. State one effect of changing the batch size from 32 to 256. [3]

### Question 7 [10]

- (a) Give the output of the following program and explain what the argument `stratify=y` achieves. [3]

```
import numpy as np
from sklearn.model_selection import train_test_split
X = np.arange(400).reshape(200, 2)
y = np.array([0] * 160 + [1] * 40)
X_tr, X_te, y_tr, y_te = train_test_split(
 X, y, test_size=0.25, stratify=y, random_state=7)
print(X_tr.shape, X_te.shape)
print(y_tr.sum(), y_te.sum())
print(round(y_te.mean(), 2))
```

- (b) A factory in Coimbatore uses a camera model to flag defective pump castings. In a test batch of 500 castings, 40 are actually defective. The model flags 50 castings as defective, of which 32 really are defective. Find the numbers of true positives, false positives, false negatives and true negatives, and hence the accuracy, precision and recall of the model. [4]
- (c) Distinguish between the parameters and the hyperparameters of a neural network, giving two examples of each. Explain why the hyperparameters should be chosen by their score on a validation set and not on the test set. [3]

## SECTION C (20 Marks)

Answer any two questions from this Section. Each question carries 10 marks. This Section covers practical uses of machine learning and computer vision.

### Question 8 [10]

- (a) A bank in Pune wants a model that flags fraudulent UPI payments as they happen. Suggest four features the model could use, and explain why the label for each past transaction is harder to obtain than it first seems. [3]
- (b) On a test set of 10,000 transactions, of which 50 are fraudulent, the model gives a fraud score for each. With a threshold of 0.5 it flags 200 transactions, of which 45 are fraudulent. With a threshold of 0.8 it flags 60 transactions, of which 36 are fraudulent. Find the precision and the recall at each threshold, and the number of genuine payments wrongly flagged at each. Recommend a threshold, with a reason. [4]
- (c) Fraudsters keep changing their methods. Explain why a model trained only on labelled past fraud may miss a new kind of fraud, and describe how the bank should keep its system effective. [3]

**Question 9 [10]**

- (a) A clinic in Hyderabad builds a k-nearest neighbour model to flag patients at risk of diabetes from two features, fasting glucose and body mass index (BMI). In its records, glucose ranges from 70 to 250 mg/dL and BMI from 15 to 40. Explain why the features must be scaled before distances are found, and find the min-max scaled values for a new patient with glucose 178 mg/dL and BMI 27.5. [3]
- (b) The scaled (glucose, BMI) values of five past patients are A (0.2, 0.3) not diabetic, B (0.7, 0.6) diabetic, C (0.5, 0.9) not diabetic, D (0.9, 0.4) diabetic and E (0.4, 0.4) not diabetic. The new patient is at (0.6, 0.5). The model gives a new patient the class held by the majority of its k nearest past patients. Find the squared Euclidean distance from the new patient to each past patient, and classify the new patient using k = 3. [4]
- (c) State the class the model would give the same new patient with k = 1 and with k = 5, and explain what goes wrong when k is made too small or too large. Why is an odd value of k usually chosen for a two-class problem? [3]

**Question 10 [10]**

- (a) Give the output of the following program, which finds objects in a black and white image, and state what cv2.boundingRect() returns. [4]

```
import numpy as np
import cv2
img = np.zeros((100, 150), np.uint8)
cv2.rectangle(img, (10, 20), (49, 59), 255, -1)
cv2.rectangle(img, (80, 30), (129, 89), 255, -1)
contours, _ = cv2.findContours(
 img, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
print(len(contours))
boxes = sorted(cv2.boundingRect(c) for c in contours)
for (x, y, w, h) in boxes:
 print(x, y, w, h, w * h)
```

- (b) A farmers' cooperative wants an app that tells fruits from vegetables in photographs taken on phones. Describe the steps needed to build and test the image classifier. [3]
- (c) A traffic camera records 25 frames per second. An object tracker follows a car whose centroid is at pixel (120, 340) in one frame and at pixel (180, 260) five frames later. One pixel corresponds to 0.05 m on the road. Find the speed of the car in metres per second and in kilometres per hour, and state why tracking is needed rather than detection alone. [3]

## ANSWER KEY AND MARKING SCHEME

A full model answer is given for every part. Marks shown in italics are the step marks an examiner would award. Every calculation in this key has been checked by machine, and every Python program quoted in a question has been run with Python 3, NumPy, Pandas, scikit-learn and OpenCV, so the outputs given are the outputs actually produced. The one exception is Keras code, which is marked on its structure, with the parameter counts checked by arithmetic. Accept any correct method that reaches the same value, and accept a value rounded sensibly. For an output question the values and their order must be right, but differences in the spacing NumPy prints inside square brackets are not penalised. For a question that asks for code, accept any sensible variable names and any equivalent library call. For an explanation, credit the idea rather than the exact wording, and accept any relevant Indian or real world example. For a drawing accept any clear and correctly labelled version.

### Question 1 [10]

- (i) (c) stemming. A stemmer cuts word endings so that forms of the same word are counted together.
- (ii) There is a strong negative linear relationship: teenagers with more screen time tend to sleep fewer hours. (It does not on its own prove that screen time causes the loss of sleep.)
- (iii) (b) 68 per cent. By the empirical rule about 68, 95 and 99.7 per cent lie within one, two and three standard deviations.
- (iv) (a) the gradient of the loss with respect to each weight, working from the output layer back towards the input, so that each weight can be adjusted by gradient descent.
- (v) The stride is the number of pixels by which the filter moves at each step as it slides across the image; a stride of 2 roughly halves the width and height of the output.
- (vi) (d) mean 0 and standard deviation 1, since each value is replaced by  $(x - \text{mean}) / \text{standard deviation}$ .
- (vii) `fit()`, as in `model.fit(X_train, y_train, ...)`.
- (viii) (b) supervised classification, because each training example carries a known label and the output is a class.
- (ix) (c) image classification, since one label is given to the whole image and no location is asked for.
- (x) `cv2.VideoCapture()` (frames are then read with its `read()` method).

*1 mark for each correct part. No mark is deducted for a wrong option, and a one-line answer that carries the correct idea gains the mark even if it is loosely worded.*

### Question 2 [10]

- (i) Mean =  $1150 / 8 = 143.75$  ms. In order: 95, 95, 95, 105, 110, 120, 130, 400, so the median =  $(105 + 110) / 2 = 107.5$  ms. Mode = 95 ms (three times). The median best describes a typical time, because the single slow result of 400 ms drags the mean above seven of the eight values, while the median is not affected by it.

*Note for the examiner: Machine checked.*

*Mean 143.75, median 107.5 and mode 95 1. Median chosen, with the outlier as the reason 1.*

- (ii) Convolution output side =  $(28 - 5) / 1 + 1 = 24$ , with one map per filter, so  $24 \times 24 \times 8$ . Max pooling halves width and height:  $12 \times 12 \times 8$ .

*24 x 24 x 8 1. 12 x 12 x 8 1.*

- (iii) Label encoding gives the states numbers 0, 1 and 2, which the network treats as quantities with an order and a size, as if Punjab were "twice" Assam, an order that does not exist. One-hot encoding makes a separate 0/1 column for each state (`state_Assam`, `state_Kerala`, `state_Punjab`), so no false order or distance is introduced.

*False order or size created by label encoding 1. One-hot gives one 0/1 column per value 1.*

- (iv) First, faults are caught before they cause a breakdown, so there are fewer sudden stoppages and less damage to costly parts such as gearboxes. Second, a turbine is serviced only when its data show a need, so money is not spent on parts and visits that were not needed, and repairs can be planned for calm, low-wind days when little power is lost. Any two such advantages.

*First advantage 1. Second advantage 1.*

- (v) In HSV the colour itself is held in one channel, hue, while brightness is held separately in value. A yellow mango in shade and in sunlight has nearly the same hue but very different B, G and R values, so a simple

hue range with cv2.inRange() picks out the colour reliably under changing light, while in BGR all three channels change together.

*Hue separates colour from brightness 1. So detection is robust to lighting, with example 1.*

### Question 3 [10]

(a) Output, as produced by running the program:

```
(3, 7)
['call', 'entry', 'free', 'me', 'now', 'recharge', 'win']
[0, 1, 2, 0, 1, 0, 1]
```

Working: CountVectorizer lower-cases the text, splits it into words and builds a vocabulary of the distinct words in alphabetical order: call, entry, free, me, now, recharge, win, which is 7 words, so X has 3 rows (messages) and 7 columns. The third message, "free free entry, win now", contains entry once, free twice, now once and win once, and none of the other words, which gives [0, 1, 2, 0, 1, 0, 1]. This is a bag-of-words vector. The word "a" is missing because the default token pattern keeps only tokens of two or more letters, so single-character words are dropped.

*Note for the examiner:* Output machine checked with scikit-learn. Accept "the default pattern ignores one-letter words" for the last mark.

*(3, 7) 1. Vocabulary in alphabetical order 1. [0, 1, 2, 0, 1, 0, 1] 1. Single-character tokens dropped by default 1.*

(b) The recogniser tags people, organisations, places, dates and amounts in every story. The app can then group all stories about the same person or company, let readers follow a city or a team, build "people in this story" panels and send alerts when a followed name appears. Difficulty: ambiguity, where the same word can be different kinds of entity; "Jordan" may be a person or a country, and "Amul" in "Amul raised milk prices" is an organisation but in "an Amul advertisement" may be read as a product. Names that are also common words, spelling variants (Bengaluru and Bangalore) and mixed Hindi-English text also cause errors.

*First way it helps the app 1. Second way it helps the app 1. One difficulty with example 1.*

(c) The labelled vectors are split into a training set and a test set. A classifier such as Naive Bayes is trained on the training set; it learns how often each word occurs in spam and in genuine messages, and labels a new message by which class makes its words more probable. It is then tested on the unseen test set with precision and recall, since a genuine message sent to the spam folder (a false positive) is costly. It must be retrained because spammers keep changing their wording and new words appear that are not in the old vocabulary, so a filter left alone slowly lets more spam through.

*Training on labelled vectors, with what the classifier learns 1. Testing on unseen data with suitable metrics 1. Reason for retraining 1.*

### Question 4 [10]

(a) 52 is one standard deviation below the mean ( $60 - 8$ ) and 76 is two above ( $60 + 16$ ). Between 52 and 60 lies half of 68 per cent = 34 per cent; between 60 and 76 lies half of 95 per cent = 47.5 per cent. So between 52 and 76:  $34 + 47.5 = 81.5$  per cent. Above 76:  $50 - 47.5 = 2.5$  per cent.

*52 and 76 as -1 and +2 standard deviations 1. 81.5 per cent 1. 2.5 per cent 1.*

(b)  $H_0$ :  $\mu = 62$  (the app makes no difference).  $H_1$ :  $\mu$  is not equal to 62 (two-tailed). Standard error =  $8 / \text{square root of } 64 = 8 / 8 = 1$ .  $z = (64.2 - 62) / 1 = 2.2$ . Since  $2.2 > 1.96$ ,  $z$  lies in the rejection region, so  $H_0$  is rejected: at the 5 per cent level the mean score of students using the app differs from (is higher than) the national mean. Because  $H_0$  has been rejected, the possible error is a Type I error, rejecting a true null hypothesis, whose probability is 5 per cent.

*Note for the examiner:* z machine checked.

*$H_0$  and  $H_1$  1. Standard error 1 and  $z = 2.2$  1. Reject  $H_0$ , with conclusion in context 1. Type I error named 1.*

(c) For a uniform distribution on 0 to 20 the probability of an interval is its length divided by 20.  $P(\text{wait} > 15) = (20 - 15) / 20 = 5 / 20 = 0.25$ .  $P(\text{wait} < 8) = 8 / 20 = 0.4$ . Mean =  $(0 + 20) / 2 = 10$  minutes.

*0.25 1. 0.4 1. Mean 10 minutes 1.*

**Question 5 [10]**

- (a) Flat P: area 850 is not more than 1,000, so go to the metro question; 1.5 km is within 2 km, so the prediction is 48 lakh rupees. Flat Q: area 1,350 is more than 1,000, so go to the bedrooms question; 3 is more than 2, so the prediction is 95 lakh rupees (distance to the metro is never asked on this branch). Difference: no one wrote these thresholds; the learning algorithm chose the questions and the cut-off values (1,000 square feet, 2 km, 2 bedrooms) from the data by finding splits that best separate cheap and expensive flats, and the leaf values are averages of past prices. Given new data, it can be retrained and the rules change.

*P: 48 lakh 1. Q: 95 lakh 1. Rules learnt from data, not programmed 1.*

- (b) Each filter covers  $3 \times 3$  pixels across all 3 colour channels, so it has  $3 \times 3 \times 3 = 27$  weights and 1 bias, 28 parameters; 16 filters give  $28 \times 16 = 448$  parameters. The flattened image has  $64 \times 64 \times 3 = 12,288$  values, so a Dense layer of 16 neurons needs  $12,288 \times 16 + 16 = 196,624$  parameters. Convolution needs far fewer because the same small filter is slid across the whole image (weight sharing) and each output looks only at a small neighbourhood (local connections). A filter that learns to detect an edge or an ear can find it anywhere in the picture, and the layers can build up from edges to shapes to whole animals.

*Note for the examiner: Counts checked by arithmetic.*

*448 with working 1. 196,624 with working 1. Weight sharing and local connections 1. Pattern found anywhere, building edges to objects 1.*

- (c) Without an activation each layer only computes a weighted sum, a linear function of its inputs, and a linear function of a linear function is still linear. However many layers are stacked, the whole network collapses to a single linear layer, so the hidden layers add nothing. A linear model can only draw straight boundaries (hyperplanes) between classes, but the pixel patterns of a handwritten 3 and 8, written in many styles, cannot be separated that way; non-linear activations let the network bend its decision boundaries and learn such complex patterns.

*Each layer linear 1. Stack collapses to one linear layer 1. Digits need non-linear boundaries 1.*

**Question 6 [10]**

- (a) Output, as produced by running the program:

[ 1. -0.5]  
[1. 0.]  
1.5

Working:  $x @ W1$  multiplies the row [1, 2] by the matrix, giving  $[1(0.5) + 2(0.25), 1(-1.0) + 2(0.5)] = [1.0, 0.0]$ ; adding  $b1$  gives  $z = [1.0, -0.5]$ .  $\text{np.maximum}(0, z)$  turns the negative value into 0, so  $h = [1.0, 0.0]$ . Then  $y = 1.0(1.0) + 0.0(-2.0) + 0.5 = 1.5$ . The line  $h = \text{np.maximum}(0, z)$  applies the activation, which is ReLU.

*Note for the examiner: Accept [1.0, -0.5] and [1.0, 0.0] written in ordinary list form.*

*z printed as [ 1. -0.5] 1. h printed as [1. 0.] 1. 1.5 1. ReLU line identified 1.*

- (b)  $\hat{y} = 1.5 \times 2 = 3$ , so  $L = (3 - 5)^2 = 4$ .  $dL/dw = 2(2)(3 - 5) = -8$ . New  $w = 1.5 - 0.05 \times (-8) = 1.5 + 0.4 = 1.9$ . New  $\hat{y} = 1.9 \times 2 = 3.8$ , new  $L = (3.8 - 5)^2 = 1.44$ , which is less than 4.

*Note for the examiner: Machine checked.*

*Loss 4 and gradient -8 1. New w = 1.9 1. New loss 1.44 1.*

- (c) Too high: each step overshoots the minimum, so the loss jumps about or grows and training may diverge. Too low: each step is tiny, so the loss falls very slowly and training may stop, at the epoch limit, before a good solution is reached. Batch size 256: there are fewer, larger batches, so fewer weight updates per epoch and each gradient is a smoother, less noisy average; each epoch runs faster on a GPU but needs more memory, and the model may generalise slightly less well. Any one effect.

*Too high: overshoot or diverge 1. Too low: very slow progress 1. One effect of larger batch 1.*

**Question 7 [10]**

- (a) Output, as produced by running the program:

(150, 2) (50, 2)  
30 10  
0.2

Working: X has 200 rows of 2 columns; test\_size=0.25 puts 50 rows in the test set and 150 in the training set. y has 40 ones out of 200, that is 20 per cent. stratify=y keeps the same proportion of each class in both sets, so the test set gets 20 per cent of 50 = 10 ones and the training set the other 30, and the mean of y\_test is 0.2. Without stratify the minority class could be under or over represented by chance.

*(150, 2) (50, 2) 1. 30 10 and 0.2 1. stratify keeps the class proportions 1.*

- (b) TP = 32. FP = 50 - 32 = 18. FN = 40 - 32 = 8. TN = 500 - 32 - 18 - 8 = 442. Accuracy =  $(32 + 442) / 500 = 474 / 500 = 0.948$ . Precision =  $32 / (32 + 18) = 32 / 50 = 0.64$ . Recall =  $32 / (32 + 8) = 32 / 40 = 0.8$ .

*Note for the examiner: Machine checked.*

*TP, FP and FN 1. TN = 442 1. Accuracy 0.948 1. Precision 0.64 and recall 0.8 1.*

- (c) Parameters are the values the network learns from the data during training, such as the weights and the biases of each layer. Hyperparameters are settings chosen by the programmer before training, which control how the learning happens, such as the learning rate, the batch size, the number of epochs and the number of hidden layers or neurons. If hyperparameters are picked by trying many settings on the test set, the choice is tuned to that test set, so its score becomes optimistic and is no longer an honest estimate of performance on new data; the validation set is used for choosing, and the test set is kept untouched for one final check.

*Parameters learnt and hyperparameters set beforehand, distinguished 1. Two examples of each 1. Reason for using the validation set, test set kept for the final check 1.*

### Question 8 [10]

- (a) Features: amount compared with the customer's usual amount; time of day; whether the payee is new to the customer; number of payments in the last hour; distance between the device location and usual location; whether the device or SIM is new; how long the account has existed. Any four. Label: a transaction is known to be fraud only after the customer complains or an investigation ends, often weeks later, and many frauds are never reported, so some fraudulent payments are wrongly labelled genuine in the training data.

*Four suitable features (two for 1 mark) 2. Label delay or unreported fraud 1.*

- (b) Threshold 0.5: precision =  $45 / 200 = 0.225$ , recall =  $45 / 50 = 0.9$ , and  $200 - 45 = 155$  genuine payments are wrongly flagged. Threshold 0.8: precision =  $36 / 60 = 0.6$ , recall =  $36 / 50 = 0.72$ , and  $60 - 36 = 24$  genuine payments are wrongly flagged. Raising the threshold improves precision but lowers recall: 9 more frauds are missed in exchange for 131 fewer false alarms. Recommendation: 0.5 if a flag only holds the payment for a quick OTP or phone check, since catching 90 per cent of fraud matters more than the extra checks; 0.8 if every flag blocks the payment and annoys genuine customers. Either recommendation earns the mark if it is justified by the costs of the two kinds of error.

*Note for the examiner: Machine checked.*

*Precision and recall at 0.5 1. Precision and recall at 0.8 1. 155 and 24 genuine payments flagged 1. Justified recommendation 1.*

- (c) A supervised model learns the patterns of the fraud it was shown, so a new trick that looks unlike past fraud may receive a low score. Customer behaviour also shifts over time (concept drift), as during festival sales, so the model slowly becomes stale. The bank should add an anomaly detection layer that flags anything unusual for that customer even without a label, have human investigators review the flagged cases and feed their verdicts back as new labels, monitor precision and recall every week, and retrain the model regularly on recent data.

*Supervised model misses unseen patterns 1. Concept drift 1. Anomaly detection, human review and regular retraining 1.*

### Question 9 [10]

- (a) kNN uses distances. Glucose varies over 180 units while BMI varies over only 25, so without scaling a small difference in glucose would swamp any difference in BMI and the model would in effect ignore BMI. Scaling puts both on 0 to 1. Glucose:  $(178 - 70) / (250 - 70) = 108 / 180 = 0.6$ . BMI:  $(27.5 - 15) / (40 - 15) = 12.5 / 25 = 0.5$ .

*Reason: unscaled glucose dominates the distance 1. Glucose 0.6 1. BMI 0.5 1.*

- (b) Squared distances from (0.6, 0.5): A,  $0.4^2 + 0.2^2 = 0.16 + 0.04 = 0.20$ ; B,  $0.1^2 + 0.1^2 = 0.02$ ; C,  $0.1^2 + 0.4^2 = 0.01 + 0.16 = 0.17$ ; D,  $0.3^2 + 0.1^2 = 0.09 + 0.01 = 0.10$ ; E,  $0.2^2 + 0.1^2 = 0.04 + 0.01 = 0.05$ . In order of nearness: B (0.02), E (0.05), D (0.10), C (0.17), A (0.20). The three nearest are B diabetic, E not diabetic and D diabetic, so the vote is 2 to 1 and the new patient is classified as at risk (diabetic).

*Note for the examiner:* Machine checked. Ranking by square-rooted distances gives the same order; accept either.

*Five squared distances 2. Three nearest B, E, D 1. Classified diabetic by a 2 to 1 vote 1.*

- (c)  $k = 1$ : the single nearest patient is B, diabetic, so the class is diabetic.  $k = 5$ : all five past patients vote, 3 not diabetic (A, C, E) to 2 diabetic (B, D), so the class is not diabetic. With  $k$  too small the answer rests on one or two neighbours and is thrown by a single unusual or mislabelled record; with  $k$  too large, distant patients outvote close ones and every new patient tends to be given the majority class of the whole data, as happens here with  $k = 5$ . An odd  $k$  cannot give a tied vote between two classes.

*Note for the examiner:* Machine checked.

*Diabetic with  $k = 1$  and not diabetic with  $k = 5$  1. Problem with too small and too large  $k$  1. Odd  $k$  avoids ties 1.*

### Question 10 [10]

- (a) Output, as produced by running the program:

```
2
10 20 40 40 1600
80 30 50 60 3000
```

Working: two separate filled white rectangles are drawn (thickness -1 means filled), so findContours with RETR\_EXTERNAL finds 2 outer contours. cv2.rectangle includes both corner pixels, so the first runs from  $x = 10$  to 49 and  $y = 20$  to 59, which is 40 pixels wide and 40 high, area 1,600; the second runs from  $x = 80$  to 129 and  $y = 30$  to 89, 50 wide and 60 high, area 3,000. Sorting the tuples puts the box with the smaller  $x$  first. cv2.boundingRect() returns  $(x, y, w, h)$ : the column and row of the top-left corner of the smallest upright rectangle round the contour, and its width and height.

*Note for the examiner:* A candidate who gives widths 39 and 49 has missed that both corners are included; award the mark for the count and for boundingRect only.

*2 1. 10 20 40 40 1600 1. 80 30 50 60 3000 1. boundingRect returns  $(x, y, w, h)$  1.*

- (b) Collect many labelled photographs of each class, taken in different light, angles and backgrounds; resize every image to one size such as 128 by 128 and scale pixel values to 0 to 1; split into training, validation and test sets; enlarge the training set by augmentation (flips, small rotations, brightness changes); train a CNN, or fine-tune a pretrained one (transfer learning), watching validation loss to avoid overfitting; finally measure accuracy, precision and recall on the untouched test set and check the mistakes before release.

*Collect, label, resize and normalise 1. Split and augment 1. Train a CNN and test on held-out data 1.*

- (c) Distance in pixels = square root of  $((180 - 120)^2 + (260 - 340)^2)$  = square root of  $(3600 + 6400)$  = square root of 10000 = 100 pixels =  $100 \times 0.05 = 5$  m. Time = 5 frames / 25 frames per second = 0.2 s. Speed =  $5 / 0.2 = 25$  m/s =  $25 \times 3.6 = 90$  km/h. Detection only finds cars in each frame separately; tracking links the detections from frame to frame and gives each car one identity, so that its path, and hence its speed, can be measured and it is counted only once.

*Note for the examiner:* Machine checked.

*100 pixels = 5 m 1. 25 m/s and 90 km/h 1. Tracking links one identity across frames 1.*