THE CHINESE UNIVERSITY OF HONG KONG, SHENZHEN

DDA 3020

MACHINE LEARNING

Assignment3 Report

Author:
Zhao Rui

Student Number:
121090820

May 21, 2023

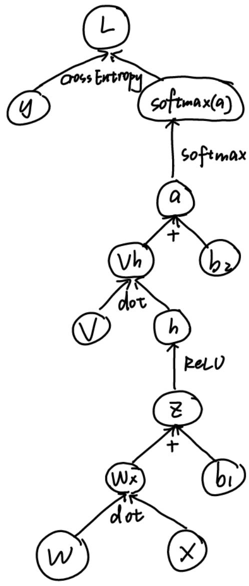
Contents

1 Written Questions

1.1 Question 1

Q1:

(a)



(b) $\nabla_V L = \frac{\partial L}{\partial a} \cdot \frac{\partial a}{\partial V} = \frac{\partial L}{\partial a} \cdot h = \frac{\partial L}{\partial a} ReLU(Wx+b)$

$\nabla_{b_2} L = \frac{\partial L}{\partial a} \cdot \frac{\partial a}{\partial b_2} = \frac{\partial L}{\partial a}$

$\nabla_W L = \frac{\partial L}{\partial a} \cdot \frac{\partial a}{\partial h} \cdot \frac{\partial h}{\partial z} \cdot \frac{\partial z}{\partial W}$

$= \frac{\partial L}{\partial a} \cdot V \cdot H(z) \cdot x$

$= \frac{\partial L}{\partial a} \cdot V \cdot H(Wx+b_1) \cdot x$

$\nabla_{b_1} L = \frac{\partial L}{\partial a} \cdot \frac{\partial a}{\partial h} \cdot \frac{\partial h}{\partial z} \cdot \frac{\partial z}{\partial b_1}$

$= \frac{\partial L}{\partial a} \cdot V \cdot H(W_x + b_1)$

1.2 Question 2

Q2:

(a) $(N+2P-F)/stride+1$

① $(32+2\times2-5)/1+1=32$

$32\times32\times10$

② after Maxpool2

$\frac{32}{2}=16$

$16\times16\times10$

③ $(16+2\times2-5)/1+1=16$

$16\times16\times10$

④ $\frac{16}{2}=8$

$8\times8\times10$

(b) first Conv5(10) layer $(5\times5\times1+1)\times10=260$ parameters

first Maxpool2 layer 0 parameters

second Cov5(10) layer $(5\times5\times10+1)\times10=2510$ parameters

second Maxpool2 layer 0 parameters

FC10 layer $(8\times8\times10+1)\times10=6410$ parameters.

1.3 Question 3

Q3:

① $H(\text{overweight}) = H(\frac{2}{3}, \frac{1}{3}) = -\frac{2}{3}\log\frac{2}{3} - \frac{1}{3}\log\frac{1}{3} = 0.9183$

$H(\text{overweight}|\text{gender}) = \frac{2}{3}H(\frac{1}{2}, \frac{1}{2}) + \frac{1}{3}H(1,0)$

$= \frac{2}{3}\times(-\frac{1}{2}\log\frac{1}{2} - \frac{1}{2}\log\frac{1}{2}) + \frac{1}{3}\times 0$

$= \frac{2}{3}$

$I(\text{overweight}; \text{gender}) = H(\text{overweight}) - H(\text{overweight}|\text{gender}) = 0.2516$ bits

$H(\text{overweight}|\text{Hyperlipidemia}) = \frac{1}{2}H(1,0) + \frac{1}{2}H(\frac{1}{3}, \frac{2}{3})$

$= \frac{1}{2}\times 0 + \frac{1}{2}(-\frac{1}{3}\log\frac{1}{3} - \frac{2}{3}\log\frac{2}{3}) = 0.4592$

$I(\text{overweight}; \text{Hyperlipidemia}) = H(\text{overweight}) - H(\text{overweight}|\text{Hyperlipidemia}) = 0.4592$ bits ✓

$H(\text{overweight}|\text{unhealthy diet}) = \frac{1}{3}H(\frac{1}{2}, \frac{1}{2}) + \frac{2}{3}H(\frac{1}{4}, \frac{3}{4})$

$= \frac{1}{3}\times 1 + \frac{2}{3}\times(-\frac{1}{4}\log\frac{1}{4} - \frac{3}{4}\log\frac{3}{4}) = 0.8742$

$I(\text{overweight}; \text{unhealthy diet}) = H(\text{overweight}) - H(\text{overweight}|\text{unhealthy diet}) = 0.0441$ bits.

$H(\text{overweight}|\text{exercises}) = \frac{2}{3}H(\frac{1}{2}, \frac{1}{2}) + \frac{1}{3}H(1,0) = \frac{2}{3}$

$I(\text{overweight}; \text{exercises}) = H(\text{overweight}) - H(\text{overweight}|\text{exercises}) = 0.2516$ bits

② $H(\text{overweight}) = -\frac{1}{3}\log(\frac{1}{3}) - \frac{2}{3}\log\frac{2}{3} = 0.9183$

$H(\text{overweight}|\text{gender}) = \frac{2}{3}H(1,0) + \frac{1}{3}H(1,0) = 0$

$I(\text{overweight}; \text{gender}) = H(\text{overweight}) - H(\text{overweight}|\text{gender}) = 0.9183$ bits

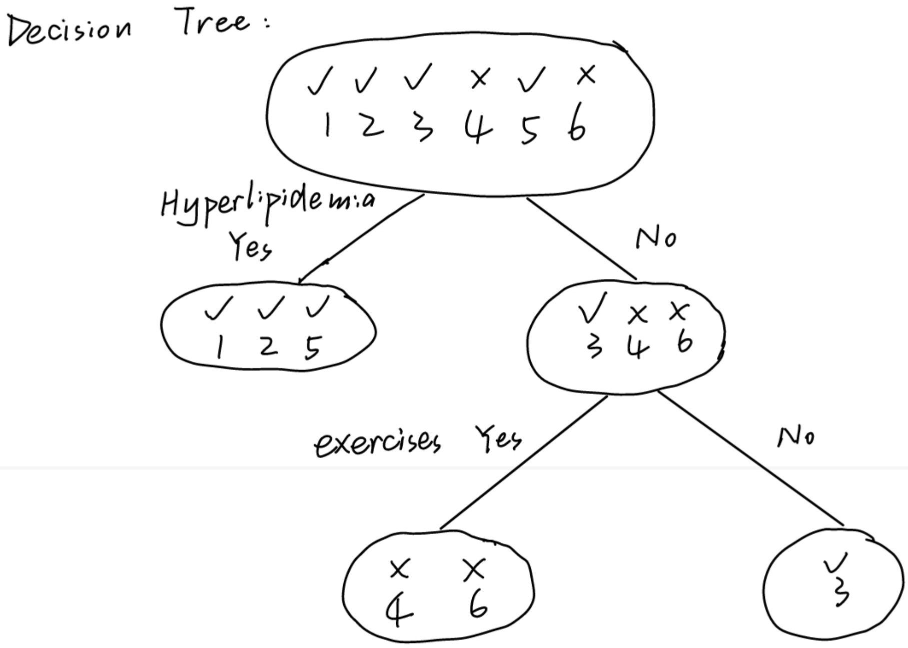
$H(\text{overweight}|\text{unhealthy diet}) = \frac{2}{3}H(\frac{1}{2}, \frac{1}{2}) + \frac{1}{3}H(1,0) = \frac{2}{3}$

$I(\text{overweight}; \text{unhealthy diet}) = H(\text{overweight}) - H(\text{overweight}|\text{unhealthy diet}) = 0.2516$ bits

$H(\text{overweight}|\text{exercises}) = \frac{2}{3}H(1,0) + \frac{1}{3}H(1,0) = 0$

$I(\text{overweight}; \text{exercises}) = H(\text{overweight}) - H(\text{overweight}|\text{exercises}) = 0.9183$ bits

At this time, we can choose both gender and exercises. Here I choose exercises.


Decision Tree:
✓ ✓ ✓ × ✓ ×
1 2 3 4 5 6
Hyperlipidemia
Yes
No
✓ ✓ ✓
1 2 5
✓ × ×
3 4 6
exercises Yes
No
× ×
4 6
✓
3

1.4 Question 4

Q4:

(a)

$$(x_1, x_2) \rightarrow f = \frac{1}{1 + e^{-(w_1 x_1 + w_2 x_2)}}$$

Positive sample:			Negative sample:		
$(5,5) \rightarrow$	0.8808	+	$(-5,1) \rightarrow$	0.0691	−
$(3,8) \rightarrow$	0.6682	+	$(2,-2) \rightarrow$	0.7685	+
$(-1,8) \rightarrow$	0.2142	−	$(-1,1) \rightarrow$	0.3543	−
$(5,4) \rightarrow$	0.8909	+	$(-5,-10) \rightarrow$	0.1824	−
$(-1,-2) \rightarrow$	0.4256	−	$(-10,-9) \rightarrow$	0.0163	−

(b)

Accuracy: $\frac{7}{10} = 70\%$

Precision: $\frac{3}{5} = 60\%$

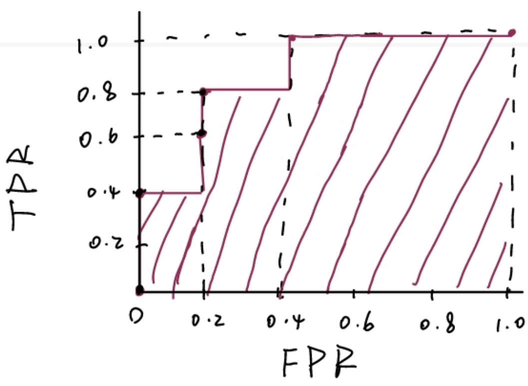
Recall: $\frac{3}{3+1} = 75\%$

F1 score: $\frac{2 \times precision \times recall}{precision + recall} = \frac{2}{3}$

Confusion matrix:

	$\bar{P}$	$\bar{N}$
P	3	2
N	1	4

(c) TPR y FPR x



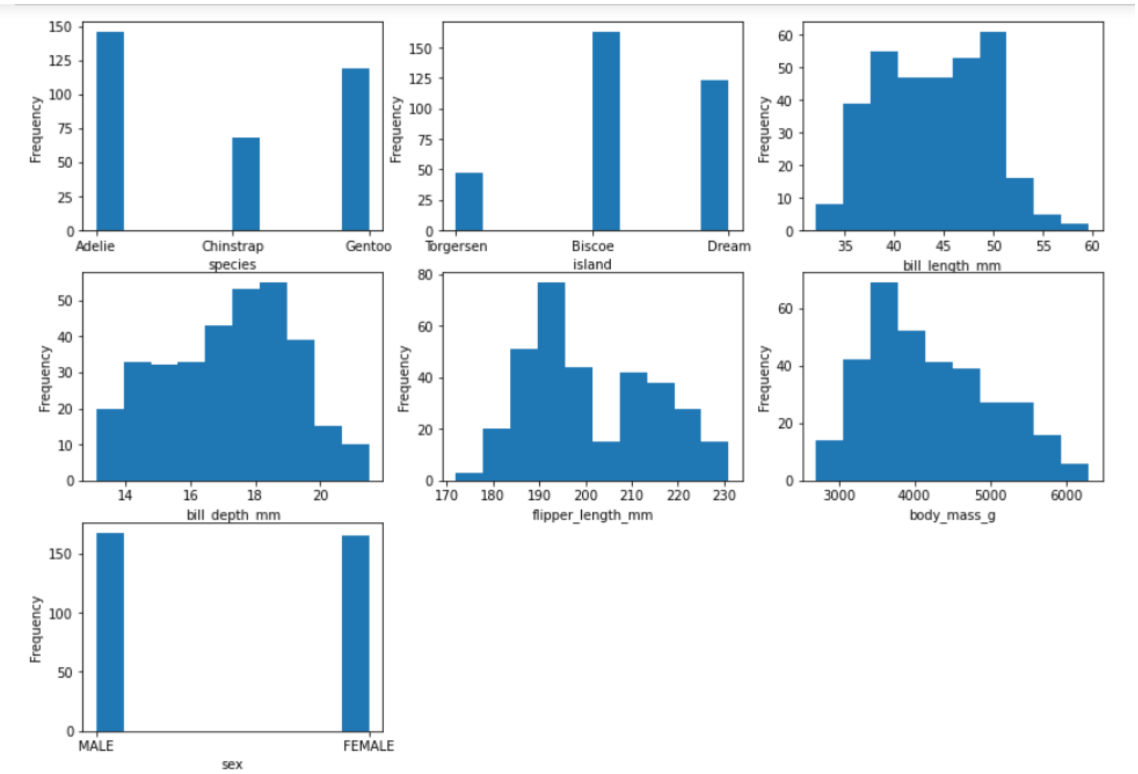
	TPR	FPR
0	1	1
0.2	1	0.4
0.4	0.8	0.2
0.6	0.6	0.2
0.8	0.4	0
1.0	0	0

(d) $AUC = 0.2 \times 0.4 + 0.8 \times 0.2 + 1.0 \times 0.6 = 0.84$

2 Programming Question

2.1 Decision Tree

First I ensure there are incomplete data points in the dataset. Then I use the dropna function to drop all the 'NAN' or 'Null'. In addition, I plot the histogram for each attribute and species of their frequency as the visualization of the statistics as follows:



Then we build a decision tree, the meaning of hyperparameters as following:

max_depth:the maximum depths of tree, here we set it as 5, 10, 15

max_leaf_nodes:the least node sizes of tree, here we set it as 50, 70, 100

And the accuracy is calculated by $\frac{correctpredict}{correctpredict+wrongpredict}$

And after training we get the accuracy and tree as follow:

when maximum depths is 5, least node sizes is 1: the train accuracy is 1.0, the test accuracy is 0.9642857142857143

when maximum depths is 5, least node sizes is 5: the train accuracy is 0.963855421686747, the test accuracy is 0.9404761904761905

when maximum depths is 5, least node sizes is 10: the train accuracy is 0.9518072289156626, the test accuracy is 0.9404761904761905

when maximum depths is 10, least node sizes is 1: the train accuracy is 1.0, the test accuracy is 0.9761904761904762

when maximum depths is 10, least node sizes is 5: the train accuracy is 0.963855421686747, the test accuracy is 0.9523809523809523

when maximum depths is 10, least node sizes is 10: the train accuracy is 0.9518072289156626, the test accuracy is 0.9404761904761905

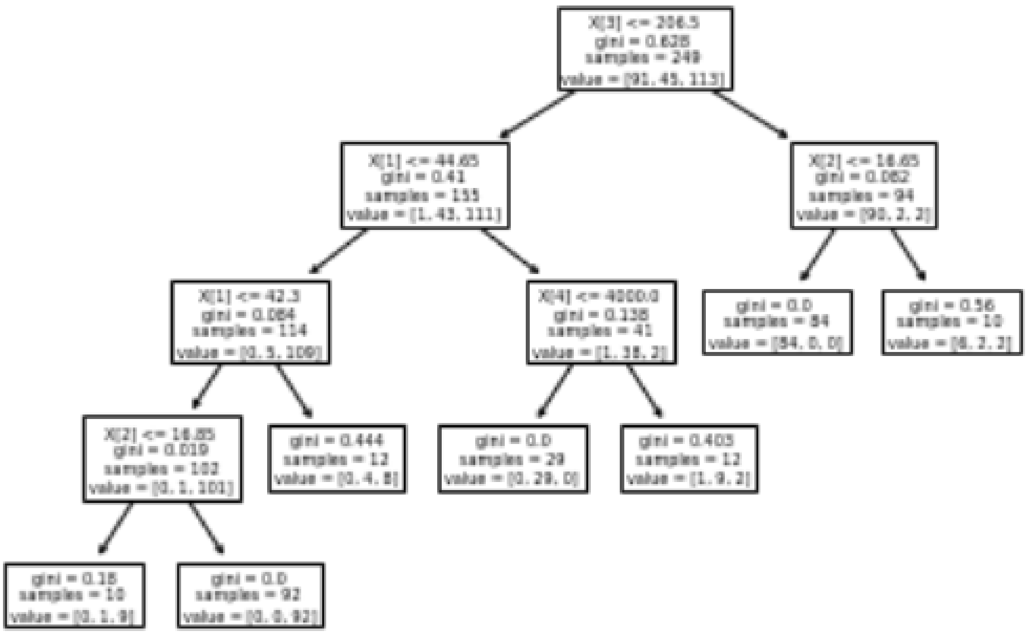
when maximum depths is 15, least node sizes is 1: the train accuracy is 1.0, the test accuracy is 0.9761904761904762

when maximum depths is 15, least node sizes is 5: the train accuracy is 0.963855421686747, the test accuracy is 0.9523809523809523

when maximum depths is 15, least node sizes is 10: the train accuracy is 0.9518072289156626, the test accuracy is 0.9404761904761905

From the result we can see that, when the least node size becomes smaller and the maximum depths becomes larger, the accuracy becomes larger.

The learned tree is plotted as follow:



For the bagging of tree, the meaning of hyperparameters as following:

max_depth:the maximum depths of tree, here we set it as 5, 10, 15

n_estimators:the number of tree, here we set it as 5, 10, 15

And after training we get the accuracy as follow:

when maximum depths is 5, the number of tree is 5: the train accuracy is 1.0, the test accuracy is 0.9642857142857143

when maximum depths is 5, the number of tree is 10: the train accuracy is 0.9959839357429718, the test accuracy is 0.9880952380952381

when maximum depths is 5, the number of tree is 15: the train accuracy is 1.0, the test accuracy is 0.9880952380952381

when maximum depths is 10, the number of tree is 5: the train accuracy is 1.0, the test accuracy is 0.9642857142857143

when maximum depths is 10, the number of tree is 10: the train accuracy is 0.9959839357429718, the test accuracy is 0.9880952380952381

when maximum depths is 10, the number of tree is 15: the train accuracy is 1.0, the test accuracy is 0.9880952380952381

when maximum depths is 15, the number of tree is 5: the train accuracy is 1.0, the test accuracy is 0.9880952380952381

when maximum depths is 15, the number of tree is 10: the train accuracy is 1.0, the test accuracy is 0.9880952380952381

when maximum depths is 15, the number of tree is 15: the train accuracy is 1.0, the test accuracy is 1.0

We can draw a conclusion that, when the number of trees becomes larger and the depths becomes larger, the accuarcy becomes larger.

For the random forests, the meaning of hyperparameters as following:

min_samples_split:the number of candidate attributes to split in every steps, here we set it as 2, 3, 4

n_estimators:the number of tree, here we set it as 5, 10, 15

And after training we get the accuracy as follow:

when m is 2, the number of tree is 5: the train accuracy is 1.0, the test accuracy is 1.0

when m is 2, the number of tree is 10: the train accuracy is 1.0, the test accuracy is 1.0

when m is 2, the number of tree is 15: the train accuracy is 1.0, the test accuracy is 1.0

when m is 3, the number of tree is 5: the train accuracy is 0.9959839357429718, the test accuracy is 0.9880952380952381

when m is 3, the number of tree is 10: the train accuracy is 0.9959839357429718, the test accuracy is 1.0

when m is 3, the number of tree is 15: the train accuracy is 1.0, the test accuracy is 0.9880952380952381

when m is 4, the number of tree is 5: the train accuracy is 1.0, the test accuracy is 0.9880952380952381

when m is 4, the number of tree is 10: the train accuracy is 1.0, the test accuracy is 0.9880952380952381

when m is 4, the number of tree is 15: the train accuracy is 0.9959839357429718, the test accuracy is 1.0

After analyzing the result, we can draw a conclusion that when m becomes larger the accuracy becomes smaller.

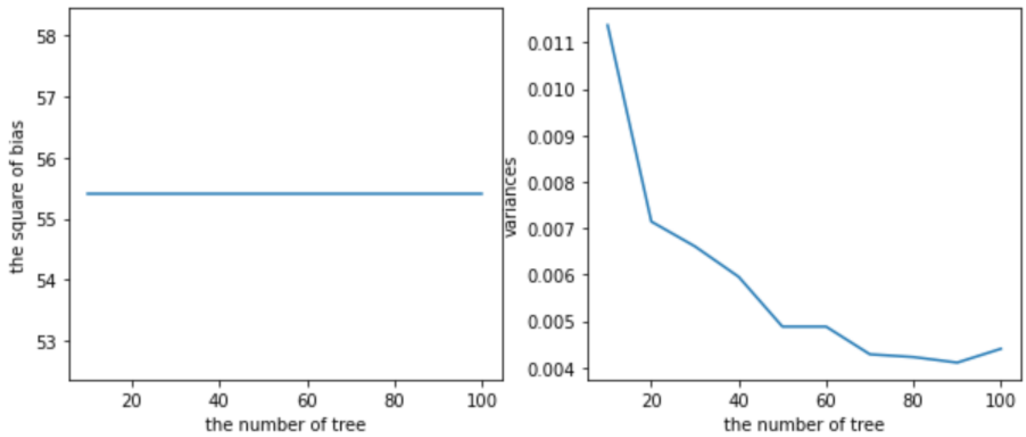
Then we use the bias_variance_decomp function to get the square of bias and variance of different numbers of trees, the formula we calculate the $bias^2$ and variance is:

$$bias^2 = E_{(x,y)}[(\bar{h}(x) - t(x))^2]$$

$$variance = E_{(x,y),D}[(h_D(x) - \bar{h}(x))^2]$$

where D is the training dataset, $h_D = A(D)$ where A is a learning algorithm, $\bar{h} = \int_D h_D p(D) dD$

then we plot the curve of $bias^2$ and variance change with the number of tree as follows:



2.2 Fashion-MNIST Recognition using sk-learn

For the random forests, the meaning of hyperparameters as following:

hidden_layer_sizes:the number of hidden nodes chosen from 50, 200, 784; besides, we need to input a tuple, for example (50,50,50) means we have 3 hidden layer with number of hidden nodes equal to 50.

solver:the optimizers chosen from adam and sgd

After training, we get the accuracy as follows:

When the number of hidden layers is 1,number of hidden nodes is 50,optimizers is adam the train accuracy is 0.8798166666666667, the test accuracy is 0.8406

When the number of hidden layers is 2,number of hidden nodes is 50,optimizers is adam the train accuracy is 0.9058833333333334, the test accuracy is 0.8658

When the number of hidden layers is 3,number of hidden nodes is 50,optimizers is adam the train accuracy is 0.9202333333333333, the test accuracy is 0.8733

When the number of hidden layers is 1,number of hidden nodes is 50,optimizers is sgd the train accuracy is 0.4347, the test accuracy is 0.429

When the number of hidden layers is 2,number of hidden nodes is 50,optimizers is sgd the train accuracy is 0.10001666666666667, the test accuracy is 0.1

When the number of hidden layers is 3,number of hidden nodes is 50,optimizers is sgd the train accuracy is 0.1, the test accuracy is 0.1

When the number of hidden layers is 1,number of hidden nodes is 200,optimizers is adam the train accuracy is 0.9131, the test accuracy is 0.8681

When the number of hidden layers is 2,number of hidden nodes is 200,optimizers is adam the train accuracy is 0.9535833333333333, the test accuracy is 0.8864

When the number of hidden layers is 3,number of hidden nodes is 200,optimizers is adam the train accuracy is 0.95885, the test accuracy is 0.8882

When the number of hidden layers is 1,number of hidden nodes is 200,optimizers is sgd the train accuracy is 0.79505, the test accuracy is 0.7888

When the number of hidden layers is 2,number of hidden nodes is 200,optimizers is sgd the train accuracy is 0.10005, the test accuracy is 0.1

When the number of hidden layers is 3,number of hidden nodes is 200,optimizers is sgd the train accuracy is 0.1, the test accuracy is 0.1

When the number of hidden layers is 1,number of hidden nodes is 784,optimizers is adam the train accuracy is 0.8952666666666667, the test accuracy is 0.8694

When the number of hidden layers is 2,number of hidden nodes is 784,optimizers is adam the train accuracy is 0.9449666666666666, the test accuracy is 0.8853

When the number of hidden layers is 3,number of hidden nodes is 784,optimizers is adam the train accuracy is 0.9801, the test accuracy is 0.8972

When the number of hidden layers is 1,number of hidden nodes is 784,optimizers is sgd the train accuracy is 0.9346166666666667, the test accuracy is 0.8602

When the number of hidden layers is 2,number of hidden nodes is 784,optimizers is sgd the train accuracy is 0.10006666666666666, the test accuracy is 0.1

When the number of hidden layers is 3,number of hidden nodes is 784,optimizers is sgd the train accuracy is 0.1, the test accuracy is 0.1

We can analyze that for sgd optimizer when the hidden layer becomes smaller and the hidden nodes become larger, the accuracy becomes larger. Besides, for adam optimizer when the hidden layer becomes larger and the hidden nodes become smaller, the accuracy becomes larger.