1: Feature Expansion

According to question:

$$f_r(x_1, x_2) = \begin{cases} +1 & 4x_1^4 + 16x_2^4 \leq r; \\ -1 & \text{otherwise} \end{cases} \tag{1}$$

Construct a function $\phi(x_1, x_2)$ to change $(x_1, x_2) = (x_1^4, x_2^4)$ will change the whole equation to linearly separable equation.
Thus we can conclude by writing the following equation:

$$\phi_r(x_1^4, x_2^4) = \begin{cases} +1 & 4x_1^4 + 16x_2^4 \leq r; \\ -1 & \text{otherwise} \end{cases} \tag{2}$$

where $w^T \phi_r(x_1^4, x_2^4) >= c$
w^T=[4,16] and c=r

2: Mistake Bound Model of Learning

(1) Here as we can observe, radius is mentioned as an integer.
total number of functions that exist is dependent on values of r.
Since we know size of concept class C = total no. of func.
So here size of concept class is 80

(2) Algorithm will only make a mistake when label is not predicted as defined by the function.
Derived Label * Actual Label <=0 then algoithm will make a mistake
or if $(x_1^2 + x_2^2 - r^2)\times\ y \leq 0$
then there is a mistake

(3) Pseudocode for given statement
If there is a mistake,
update r with integer($\sqrt{x^2 + y^2 + 1}$)
Thus the (x_1, x_2) pair gets the required label.

(4) Mistake driven learning algo to learn the function
1. (x_1, x_2) pair (taken from domain of previous answers).
2. Lets consider radius = r = 1
3. For the pair input taken in step 1, if the label output $(x_1^2 + x_2^2 - r^2)$ is positive, then we can say that the pair (x_1, x_2) lies inside the range (circle).
4. Get next label
5. Get the value of (new or derived label) *(original or old label)
6. If the value in previous step is positive, then there is no mistake.
7. Repeat the above steps for further values.

As we see the maximum no. of mistakes that algo can make is 79.
This situation will occur only when sorted input of co-ordinates is provided to the algorithm.
Value of r will get updated on every single step in the above assumed case.

(5a) By storing the labels and radius, we can overcome this problem

(5b) If most number of the functions present in concept class will predict a label (new label) that is different from the original label, we can state that a mistake has occured.

(5c) Pseudocode 1. Let the radius be r=1.
2. (x_1, x_2) is the input pair.
3. The total number of functins contained in concept class is $(x_1^2 + x_2^2 = r^2)$
4. For a pair of valid input we need to verify if positive or negative label is predicted.Most number of functions with similar output is taken into consideration and derived label (new label) is considered as the majority value.
5. If predicted label is not matching with the original label or old label (y), the func. that predicted a wrong label are removed.
6. Repeat the above steps and stop when only a single function is left in concept class.

Mistake bound value = log(80)

3: Perceptron Algorithm and its Variants - EXPERIMENTS

(1) New Weight Vector
$[\ 0,\ \ 1,\ \ 0,\ \ -1,\ \ 2\ ]$
4 is the number of mistakes

(2) Weight vector is random ranging from values -2 and 2.
Bias is also random ranging from -2 and 2.
Learning rates taken = 1 and 0.1
FOR LEARNING RATE 1
Mistakes SIMPLE PERCEPTRON = 1445
Mistakes MARGIN PERCEPTRON (1) = 1520
SIMPLE PERCEPTRON

TEST ON TRAIN SET

ACCURACY
76.78

TEST ON TEST SET

ACCURACY
77.76

MARGIN PERCEPTRON

TEST ON TRAIN SET

ACCURACY
81.96

TEST ON TEST SET

ACCURACY
81.74

FOR LEARNING RATE 0.1
Mistakes SIMPLE PERCEPTRON = 1414
Mistakes MARGIN PERCEPTRON (1) = 1957
SIMPLE PERCEPTRON

TEST ON TRAIN SET

ACCURACY
71.42

TEST ON TEST SET

ACCURACY
70.82

MARGIN PERCEPTRON

TEST ON TRAIN SET

ACCURACY
77.90

TEST ON TEST SET

ACCURACY
77.01

(3) Weight vector is random ranging from values -2 and 2.
Bias is also random ranging from -2 and 2.
Below results are reported after shuffling the data
SINCE training is done on same "a5a.train" file, number of mistakes while training are same
TEST ON TRAIN SET

MARGIN	**MISTAKES WHILE TRAINING**	**LEARNING RATE**	**EPOCH**	**ACCURACY**
0	7223	1	3	74.38
0	12147	1	5	72.70
0	7079	0.1	3	75.41
0	11741	0.1	5	71.53
0	7023	0.01	3	72.13
0	11602	0.01	5	72.84

TEST ON TEST SET

MARGIN	**MISTAKES WHILE TRAINING**	**LEARNING RATE**	**EPOCH**	**ACCURACY**
1	7223	1	3	73.88
1	12147	1	5	71.57
1	7079	0.1	3	75.77
1	11741	0.1	5	70.05
1	7023	0.01	3	70.45
1	11602	0.01	5	71.94

As we see due to random initialization of weight vector and bias and due to shuffling of data , the results vary by a large margin everytime we run the program. The results are however similar for epoch 3 and 5 in every run.

(4) Weight vector is random ranging from values -2 and 2.
Bias is also random ranging from -2 and 2.
NON - SHUFFLED DATA
TEST ON TRAIN SET

MARGIN	**MISTAKES WHILE TRAINING**	**LEARNING RATE**	**EPOCH**	**ACCURACY**
0	4466	1	3	76.50
0	7463	1	5	77.23

TEST ON TEST SET

MARGIN	**MISTAKES WHILE TRAINING**	**LEARNING RATE**	**EPOCH**	**ACCURACY**
0	4459	1	3	74.73
0	7406	1	5	75.61

Weight vector is random ranging from values -2 and 2.
Bias is also random ranging from -2 and 2.
SHUFFLED DATA
TEST ON TRAIN SET

MARGIN	**MISTAKES WHILE TRAINING**	**LEARNING RATE**	**EPOCH**	**ACCURACY**
1	11636	1	3	69.19
1	12076	1	5	68.78

TEST ON TEST SET

MARGIN	**MISTAKES WHILE TRAINING**	**LEARNING RATE**	**EPOCH**	**ACCURACY**
1	12076	1	3	68.83
1	20187	1	5	68.70