**Assignment 2** net-id:vxc140830

**Problem 1**.

1. a. The code for the Primal SVM and the accuracy calculation is as below.

function z = plotsvmv(c)

data=importdata('/Users/veenac/Downloads/spambase\_train.data');

x=data(:,1:57)

y=data(:,58);

n=57;

rows=2760;

for i=1:rows

if y(i,1)==0

y(i,1)=-1;

end

end

H=diag([ones(1,n),zeros(1,n+1)]);

f=[zeros(1,n+1),c\*ones(1,n)];

b=-ones(rows,1);

xyy= diag(y)\*[x,ones(rows,1)];

part2=ones(rows,n);

A = -[xyy,part2];

z=quadprog(H,f,A,b);

function acc = accuracy(c)

z=plotsvmv(c);

w=z(1:57,:);

b=z(58,:);

si=z(59:115,:);

data=importdata('/Users/veenac/Downloads/spambase\_test.data');

x=data(:,1:57);

y=data(:,58);

n=57;

rows=921;

count=0;

for i=1:rows

yresult = w'\*x(i,:)'+b;

if((yresult<0 && y(i,:)==0) || (yresult>0 && y(i,:)==1) )

count=count+1;

end

end

acc=(count/rows)\*100;

b. On the training data, the accuracy values are as follows

|  |  |
| --- | --- |
| C | Accuracy |
| 1 | 40.3261 |
| 10 | 89.0942 |
| 100 | 88.5145 |
| 1000 | 40.2899 |
| 10000 | 86.9928 |

c. On the validation set, the accuracy values are as follows:

|  |  |
| --- | --- |
| C | Accuracy |
| 1 | 40.5435 |
| 10 | 89.3478 |
| 100 | 88.5870 |
| 1000 | 40.5435 |
| 10000 | 87.8261 |

From the validation set values, we can take c=10.

d. On the test set, the accuracy values for c=10 is 90.0109.

2. a. The code for the dual SVM and the accuracy calculation is as below.

function [w,b] = plotdualsvmv(c,sigma)

data=importdata('/Users/veenac/Downloads/spambase\_train.data');

x=data(:,1:57);

y=data(:,58);

n=57;

rows=2760;

for i=1:rows

if y(i,1)==0

y(i,1)=-1;

end

end

H=ones(rows,rows);

f=-ones(1,rows);

for k=1:rows

for j=1:rows

kernal=gaussianKernel(x(k,:),x(j,:),sigma);

H(k,j)=kernal\*y(k,:)\*y(j,:);

end

end

Aineq=y';

bineq=0;

lb=zeros(rows,1);

ub=c\*ones(rows,1);

%options = optimoset('TolFun',1e-12);

z=quadprog(H,f,[],[],Aineq,bineq,lb,ub);

w=ones(n,1);

sum=0;

for k=1:n

for r=1:rows

sum=sum+(z(r,:)\*y(r,:)\*x(r,k));

end

w(k,:)=sum;

sum=0;

end

for i=1:rows

if(z(i,:)>0)

b=1-w\*x(i,:);

break;

end

end

end

function acc = accuracydual(c,sigma)

[w,b]=plotdualsvmv(c,sigma);

%train=importdata('C:\Users\vxc140830\Downloads\spambase\_train.data');

validation=importdata('/Users/veenac/Downloads/spambase\_validation.data');

test=importdata('/Users/veenac/Downloads/spambase\_test.data');

data=validation;

x=data(:,1:57);

y=data(:,58);

rows=920;

count=0;

for i=1:rows

yresult = w'\*x(i,:)'+b;

if((yresult<0 && y(i,:)==0) || (yresult>0 && y(i,:)==1) )

count=count+1;

end

end

acc=(count/rows)\*100;

end

function sim = gaussianKernel(x1, x2, sigma)

%RBFKERNEL returns a radial basis function kernel between x1 and x2

% Ensure that x1 and x2 are column vectors

x1 = x1(:); x2 = x2(:);

% sim = gaussianKernel(x1, x2) returns a gaussian kernel between x1 and x2

% and returns the value in sim

xny = x1-x2;

Normxny = xny'\*xny;

sim = exp(-Normxny/(2\*(sigma)^2));

end

b. On the training data, the accuracy values are as follows

|  |  |  |
| --- | --- | --- |
| C | Sigma | Accuracy |
| 1 | 0.001 | 69.0217 |
| 1 | 0.01 | 69.0217 |
| 1 | 0.1 | 69.0217 |
| 1 | 1 | 69.0217 |
| 1 | 10 | 68.8043 |
| 1 | 100 | 68.8043 |
| 10 | 0.001 | 69.0217 |
| 10 | 0.01 | 69.0217 |
| 10 | 0.1 | 66.8478 |
| 10 | 1 | 69.0217 |
| 10 | 10 | 68.8043 |
| 10 | 100 | 68.8043 |
| 100 | 0.001 | 69.0217 |
| 100 | 0.01 | 69.0217 |
| 100 | 0.1 | 69.0217 |
| 100 | 1 | 69.0217 |
| 100 | 10 | 68.8043 |
| 100 | 100 | 68.8043 |
| 1000 | 0.001 | 69.0217 |
| 1000 | 0.01 | 69.0217 |
| 1000 | 0.1 | 69.0217 |
| 1000 | 1 | 69.0217 |
| 1000 | 10 | 68.8043 |
| 1000 | 100 | 68.8043 |
| 10000 | 0.001 | 69.0217 |
| 10000 | 0.01 | 69.0217 |
| 10000 | 0.1 | 69.0217 |
| 10000 | 1 | 69.0217 |
| 10000 | 10 | 68.8043 |
| 10000 | 100 | 68.8043 |

c. On the validation set, the accuracy values are as below.

|  |  |  |
| --- | --- | --- |
| C | Sigma | Accuracy |
| 1 | 0.001 | 69.0217 |
| 1 | 0.01 | 69.0217 |
| 1 | 0.1 | 69.0217 |
| 1 | 1 | 69.0217 |
| 1 | 10 | 68.8043 |
| 1 | 100 | 68.8043 |
| 10 | 0.001 | 69.0217 |
| 10 | 0.01 | 69.0217 |
| 10 | 0.1 | 66.8478 |
| 10 | 1 | 69.0217 |
| 10 | 10 | 68.8043 |
| 10 | 100 | 68.8043 |
| 100 | 0.001 | 68.8043 |
| 100 | 0.01 | 66.8478 |
| 100 | 0.1 | 66.8478 |
| 100 | 1 | 66.8478 |
| 100 | 10 | 66.8478 |
| 100 | 100 | 66.8478 |
| 1000 | 0.001 | 66.8478 |
| 1000 | 0.01 | 66.8478 |
| 1000 | 0.1 | 66.8478 |
| 1000 | 1 | 66.8478 |
| 1000 | 10 | 66.6304 |
| 1000 | 100 | 66.7391 |
| 10000 | 0.001 | 66.8478 |
| 10000 | 0.01 | 66.8478 |
| 10000 | 0.1 | 66.8478 |
| 10000 | 1 | 66.8478 |
| 10000 | 10 | 66.8478 |
| 10000 | 100 | 66.6304 |

C has been chosen as: c=1

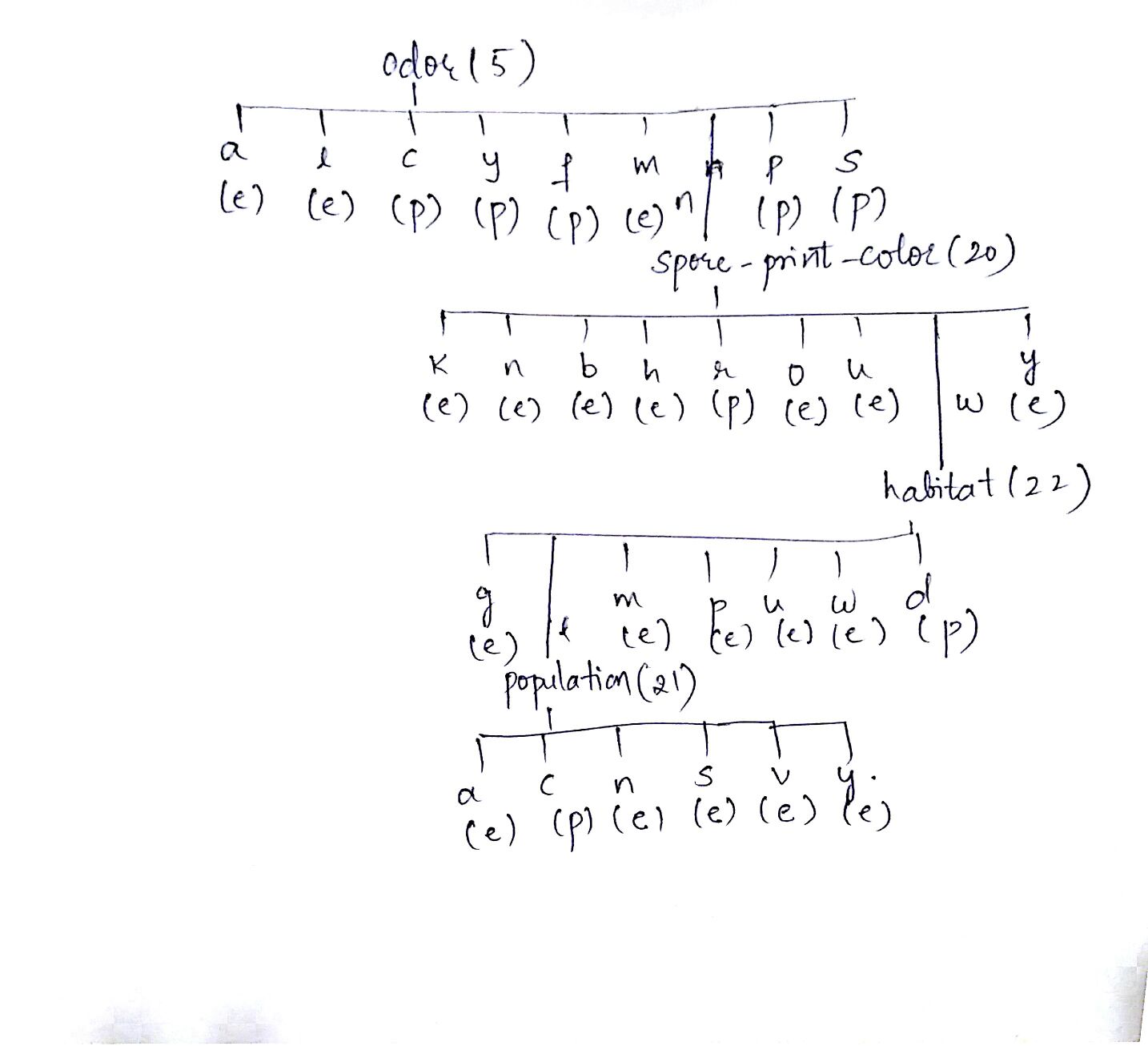
d. Accuracy on test data, with c=1, is 69.4897

3. Primal SVM should be chosen for Spam email identification, as it is giving more accurate results.

**Problem 2:**

1.The Code for building decision tree is attached to this assignment.

The decision tree obtained is as shown below.



2. The number of nodes in the decision tree (size) is 32.

3.The depth of the decision tree is 3.

4.The decision tree is 100% accurate on the training data

5. On the test data, the obtained decision tree gives 98.64% accuracy.

6. Decision tree has only a few attributes with a high information gain. It means, these are the most informative attributes out of all the given attributes. Thus Decision tree reduces the dependency to only a fewer attributes, in our case it is 4 out of 22.

Hence, using decision trees, we can form a simple set of rules to determine whether the mushroom is edible or not.

7. The decision tree depends on the training and test data splits. This is because, we are training the decision tree based on the training data, so the decision tree mainly depends on the training data. If the training data is small, it may not sufficient to determine correct information gain as the data may not include all the possibilities of the corresponding attributes, thus reducing the accuracy and the classification may not occur correctly. Hence, the training data should be as large as possible.

**Problem 3:**

1. In the one-nearest neighbor algorithm, we decide the label of the test data depending on the nearest point to the data. If both the training sets result in a positive label, it implies that in both of the training sets, the nearest point to the data point is classified as positive.

Let d1- be the data point nearest after d1+(which is nearest) to the data point x. And let d2- be the data point nearest after d2+ to x. After combining S1 and S2, any one of d1+ or d2+ will be nearest to x.

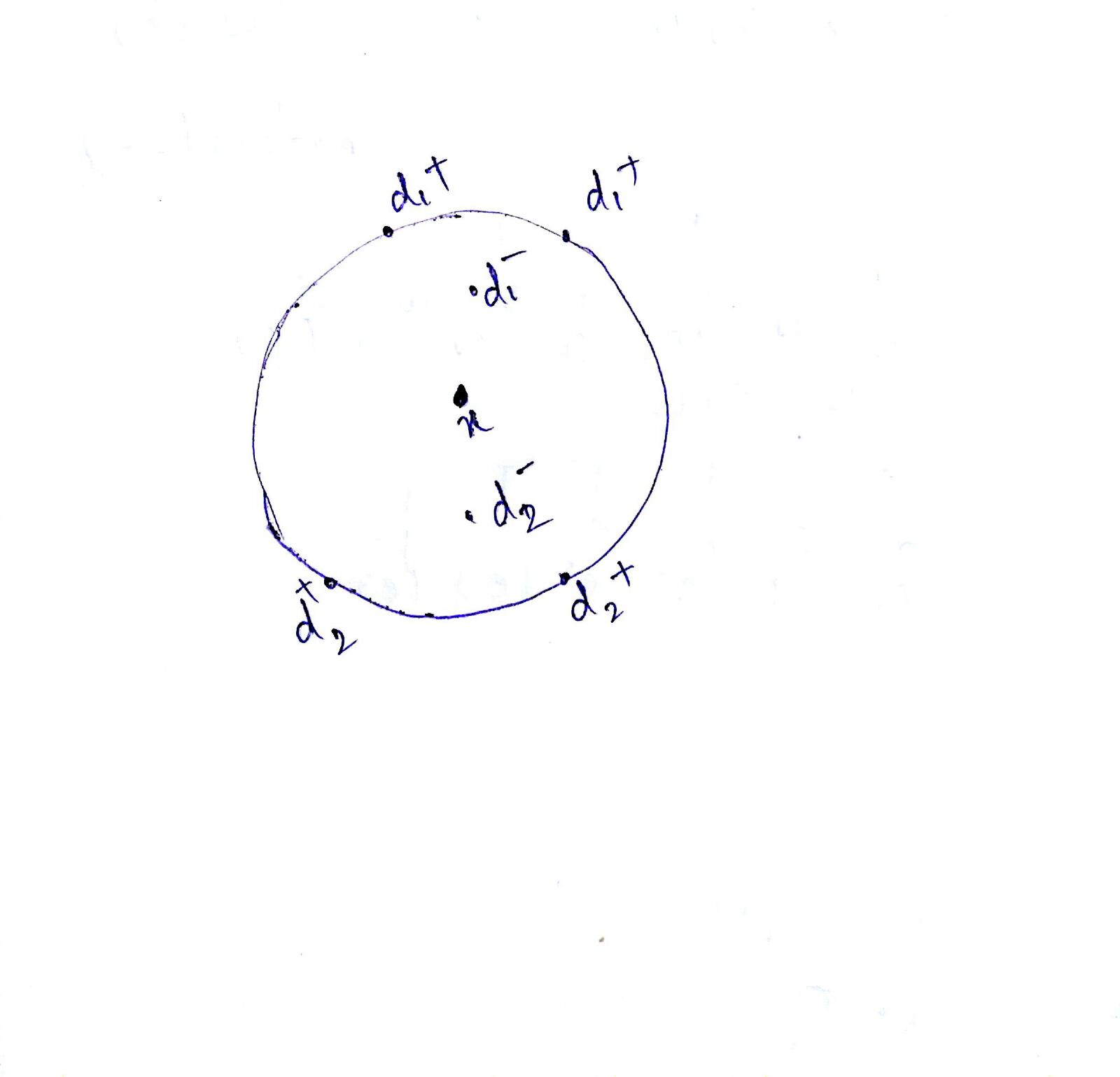
D1+ < d1-

D2+ < d2-

From the above any one of D1+ or D2+ should be nearest to the test data point.

Hence, one-nearest neighbor algorithm will label x as positive.

2.



Consider the above example. Let there be d1+’s and d1- belong to set1 and d2+’s and d2- belong to set2.

From the diagram we can say that, for set1 alone, x is labeled as positive, since there are two positive points near it. And for the set2 too, x will labeled as positive. But for S1 union S2, x will be labeled as negative.

Thus in case of three-nearest neighbor, S1 union S2 is not labeled as positive.