CIS 472/572 Homework #3 (Written)

Mamtaj Akter

TOTAL POINTS

125.5 / 160

QUESTION 1

Question 1 30 pts

1.1 Part a 0 / 10

+ **4 pts** Adequate description for the choice of normalization and weighting.
+ **6 pts** Normalization correct
+ **4 pts** Training data normalization is correct. Sample data normalization is wrong or missing.
+ **4 pts** Workings for the normalization not shown
+ **5 pts** Training data normalization is correct. Test data normalization not adequately shown.
+ **0 pts** No description for the choice of normalization and weighting.

✓ + **0 pts** **Wrong normalization.**

+ **0 pts** One choice of normalization is to subtract the mean and then divide by the standard deviation. Another is 0-1 scaling: (V - min V)/(max V - min V).
+ **1 pts** Wrong weight decisions. For example, Age of history is an important factor there. Nothing is absolute though, but it is more likely to have a better score if your history is longer.
+ **3 pts** Payment History column has saturated.
+ **0 pts** Result of the normalization not shown
+ **5 pts** Normalization only partially correct

✓ + **0 pts** **Adequate description for the choice of normalization and weighting not given.**

+ **2 pts** For a and b normalizing should be done using all 10 data points.
+ **2 pts** Description for the choice of normalization and weighting inadequate.

1.2 Part b 7.5 / 10

+ **10 pts** Correct

✓ + **7.5 pts** **Answer is correct. But not enough work shown.**

+ **5 pts** Answer is correct. But no work is shown.
+ **4 pts** Answer is correct. Not enough work shown. Examples 1-10 should be used here.
+ **0 pts** Calculations wrong/missing. Answers wrong.
+ **2 pts** Assignment wrong/missing. Assignment should be Bad, Good, Good.
+ **1 pts** Assignment not shown. Assignment should be Bad, Good, Good.

1.3 Part c 8 / 10

✓ - **2 pts** **Calculations slightly off.**
k=3 should have given
P1= Bad, P2=Good, P3 =Good

QUESTION 2

2 Question 2 40 / 40

✓ + **15 pts** **(a) Correct**

✓ + **15 pts** **(b) Correct**

✓ + **10 pts** **(c) Correct/Adequate**

+ **0 pts** (a) Accuracy values not included.
+ **5 pts** (a) Accuracy should have been decreasing.
+ **10 pts** (b) The non uniformity of the x axis scale of the graph obscures the expected graph shape.
+ **5 pts** (b) Accuracy should have been decreasing.
+ **0 pts** (b) Graph not included.
+ **0 pts** (c) Code not included.

QUESTION 3

Question 3 30 pts

3.1 Part a 15 / 15

✓ + **15 pts** **Correct**

+ **13 pts** Your structure is correct. But for the dataset we expected you to declare data points.
+ **10 pts** You have mixed up the two distances. The

height of the box should be smaller than the width and length.

+ **5 pts** This data set will not give the expected result.

+ **5 pts** In your case there is no way to get a 100% for 3NN in the case of - (negative) points.

+ **2 pts** Your observation is correct. But we asked for you to come up with a data set.

+ **0 pts** Assume a plane and put some points belonging to one class on it. Then suppose a copy of the original plane with the exact same points but with different class label. If the distance of two plains is smaller than the shortest distance of points in one plane, then we found the solution.

+ **0 pts** No labels given, thus the answer is unacceptable.

+ **3 pts** The question asks for the points themselves not the distances.

+ **2 pts** 3-nn of point 2,2 is (-,++) which classifies it incorrectly.

+ **3 pts** You need at least 6 points.

+ **2 pts** The first point is classified incorrectly, and you cannot have infinite data.

+ **0 pts** Click here to replace this description.

+ **10 pts** Click here to replace this description.

3.2 Part b 15 / 15

✓ + **15 pts** **Correct**

+ **2 pts** Not answered.

+ **0 pts** The solution is a dataset in which all the points have similar labels.

+ **0 pts** Click here to replace this description.

+ **2 pts** incorrect answer.

QUESTION 4

4 Question 4 10 / 30

✓ - **20 pts** **Substantial issues: Clarity, detail, correctness**

Perceptron predictions are +1/-1, not continuous as in logistic regression. Initial weights should all be zero. Perceptron update occurs when prediction is incorrect or there's a tie.

QUESTION 5

5 Question 5 30 / 30

✓ + **30 pts** **Correct/Adequate**

+ **30 pts** Undergrad submission. Question N/A

+ **20 pts** Inadequate

+ **10 pts** Bonus: Both sides proved.

1. A: After Normalizing the dataset:

ID	Total Accounts	Utilization	Payment History	Age of History (days)	Inquiries	Label	Distance from P1	Distance from P2	Distance from P3
1	0.08	0.15	1	0.1	0.5	GOOD	0.871435597	0.120519708	0.224944438
2	0.15	0.19	0.9	0.25	0.8	BAD	0.546443044	0.47013296	0.240416306
3	0.1	0.35	1	0.05	1	BAD	0.49244289	0.650326841	0.520672642
4	0.11	0.4	0.95	0.2	0.6	BAD	0.66565757	0.392969464	0.291376046
5	0.12	0.1	0.99	0.3	0.6	GOOD	0.746324326	0.318943569	0.042426407
6	0.18	0.15	1	0.2	0.5	GOOD	0.827888881	0.20862646	0.169115345
7	0.03	0.21	1	0.15	0.7	BAD	0.680441033	0.337083076	0.206397674
8	0.14	0.04	1	0.35	0.5	GOOD	0.851586754	0.322838969	0.152315462
9	0.13	0.05	1	0.3	0.3	GOOD	1.02464628	0.273906918	0.313209195
10	0.06	0.25	0.94	0.28	0.9	BAD	0.450111097	0.571948424	0.328937684

B:

ID	Total Accounts	Utilization	Payment History	Age of History (days)	Inquiries	Label	MIN DISTANCE	
	0.2	0.5	0.9	0.45	1.2	P1	0.450111097	BAD
	0.08	0.1	1	0.055	0.4	P2	0.120519708	GOOD
	0.09	0.13	0.99	0.3	0.6	P3	0.042426407	GOOD

1.1 Part a **0 / 10**

+ **4 pts** Adequate description for the choice of normalization and weighting.

+ **6 pts** Normalization correct

+ **4 pts** Training data normalization is correct.
Sample data normalization is wrong or missing.

+ **4 pts** Workings for the normalization not shown

+ **5 pts** Training data normalization is correct.
Test data normalization not adequately shown.

+ **0 pts** No description for the choice of normalization and weighting.

✓ + **0 pts** **Wrong normalization.**

+ **0 pts** One choice of normalization is to subtract the mean and then divide by the standard deviation. Another is 0-1 scaling: (V - min V)/(max V - min V).

+ **1 pts** Wrong weight decisions. For example, Age of history is an important factor there. Nothing is absolute though, but it is more likely to have a better score if your history is longer.

+ **3 pts** Payment History column has saturated.

+ **0 pts** Result of the normalization not shown

+ **5 pts** Normalization only partially correct

✓ + **0 pts** **Adequate description for the choice of normalization and weighting not given.**

+ **2 pts** For a and b normalizing should be done using all 10 data points.

+ **2 pts** Description for the choice of normalization and weighting inadequate.

1. A: After Normalizing the dataset:

ID	Total Accounts	Utilization	Payment History	Age of History (days)	Inquiries	Label	Distance from P1	Distance from P2	Distance from P3
1	0.08	0.15	1	0.1	0.5	GOOD	0.871435597	0.120519708	0.224944438
2	0.15	0.19	0.9	0.25	0.8	BAD	0.546443044	0.47013296	0.240416306
3	0.1	0.35	1	0.05	1	BAD	0.49244289	0.650326841	0.520672642
4	0.11	0.4	0.95	0.2	0.6	BAD	0.66565757	0.392969464	0.291376046
5	0.12	0.1	0.99	0.3	0.6	GOOD	0.746324326	0.318943569	0.042426407
6	0.18	0.15	1	0.2	0.5	GOOD	0.827888881	0.20862646	0.169115345
7	0.03	0.21	1	0.15	0.7	BAD	0.680441033	0.337083076	0.206397674
8	0.14	0.04	1	0.35	0.5	GOOD	0.851586754	0.322838969	0.152315462
9	0.13	0.05	1	0.3	0.3	GOOD	1.02464628	0.273906918	0.313209195
10	0.06	0.25	0.94	0.28	0.9	BAD	0.450111097	0.571948424	0.328937684

B:

ID	Total Accounts	Utilization	Payment History	Age of History (days)	Inquiries	Label	MIN DISTANCE	
	0.2	0.5	0.9	0.45	1.2	P1	0.450111097	BAD
	0.08	0.1	1	0.055	0.4	P2	0.120519708	GOOD
	0.09	0.13	0.99	0.3	0.6	P3	0.042426407	GOOD

1.2 Part b 7.5 / 10

+ **10 pts** Correct

✓ + **7.5 pts** **Answer is correct. But not enough work shown.**

+ **5 pts** Answer is correct. But no work is shown.

+ **4 pts** Answer is correct. Not enough work shown.

Examples 1-10 should be used here.

+ **0 pts** Calculations wrong/missing.

Answers wrong.

+ **2 pts** Assignment wrong/missing.

Assignment should be Bad, Good, Good.

+ **1 pts** Assignment not shown.

Assignment should be Bad, Good, Good.

C.

ID	Total Accounts	Utilization	Payment History	Age of History (days)	Inquiries	Label	DISTANCE FROM ID7 (BAD)	Distance from ID8 (GOOD)	Distance from ID9 (GOOD)	Distance from ID10 (BAD)
1	0.08	0.15	1	0.1	0.5	GOOD	0.220454077	0.279642629	0.304138127	0.454312668
2	0.15	0.19	0.9	0.25	0.8	BAD	0.211660105	0.364142829	0.531507291	0.155563492
3	0.1	0.35	1	0.05	1	BAD	0.352845575	0.661588996	0.802122185	0.279463772
4	0.11	0.4	0.95	0.2	0.6	BAD	0.239791576	0.406816912	0.474763099	0.348568501
5	0.12	0.1	0.99	0.3	0.6	GOOD	0.229782506	0.128840987	0.304466747	0.344963766
6	0.18	0.15	1	0.2	0.5	GOOD	0.261916017	0.190262976	0.25	0.440908154

For k=1, its accuracy is not 1 as ID7=(ID1=GOOD)=BAD, ..
For k=2, ID7=(GOOD,BAD), it's a tie, so lets take k=3
For k=3, its accuracy is not 1 as ID7=(ID1=GOOD,ID2=BAD,ID6=GOOD)=BAD, …….
For k=4, ID7==(ID1=GOOD,ID2=BAD,ID5=GOOD,ID4=BAD)=BAD, it's a tie, so lets take k=5
For k=5, its accuracy is still not 1 as ID7=(ID1=GOOD,ID2=BAD,ID6=GOOD,ID5=GOOD, ID4=BAD)=BAD…
For k=6 is the best

1.3 Part c 8 / 10

✓ - 2 pts Calculations slightly off.

k=3 should have given

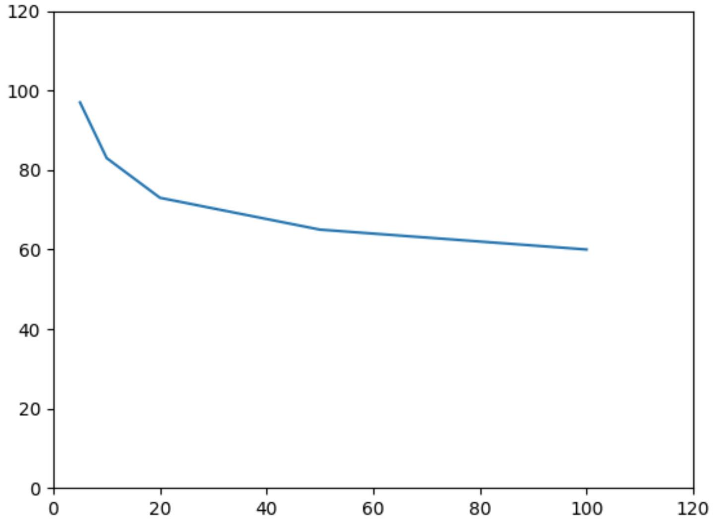
P1= Bad, P2=Good, P3 =Good

2.

a)

Dimension	Average Accuracies
5	97
10	83
20	73
50	65
100	60

b)



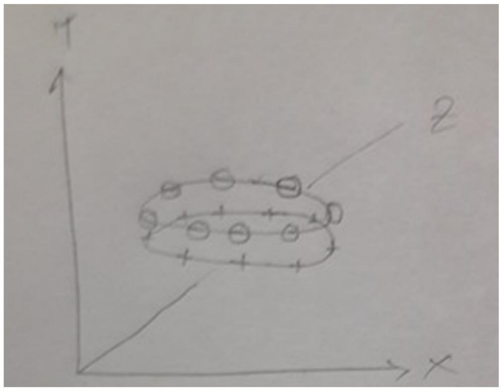
c)

```
import numpy as np
import pandas as pd
import matplotlib
import matplotlib.pyplot as plt
import os
import glob
import random
import itertools
from scipy.spatial import distance
from random import randint

def random_gen(low, high):
  while True:
    yield random.randrange(low, high)

def getColumn(lst,i):
        return [column[i] for column in lst]

def makeDataSet(yz):
        train_data=[]
        randomRows=[]
        for i in xrange(100):
                train_data.append([])
                for j in xrange(100):
                        train_data[i].append(1)
        i=0
        j=0
        for j in range (len(train_data[0])):#columns
                if j==0:
                        for i in range(len(train_data)):
```

```
                if i<50:
                    train_data[i][j]=0
                else:
                    train_data[i][j]=1
        else:
            for i in range(len(train_data)):
                train_data[i][j]=randint(0, 1)
    return train_data

def makeDistances(train_data,test_data):
    distances=[]
    i=0
    j=0
    for i in range (len(test_data)):
        distances.append([])
        for j in range (len(train_data)):
            distances[i].append( distance.euclidean(test_data[i],train_data[j]))
    return distances

def getminimumDistanceIndexes(distances):
    return np.argmin(distances, axis=0)

counter=0
accuracies=[]
for k in range(10):
    train_data=[]
    test_data=[]
    train_data=makeDataSet ("Nothing")
    test_data=makeDataSet ("Nothing")
    distances=[]
    distances=makeDistances(train_data,test_data)
```

```
        minimumDistanceIndexes=[]
        minimumDistanceIndexes=getminimumDistanceIndexes(distances)
        counter=0
        i=0
        j=0
        for i in range(len(minimumDistanceIndexes)):
                j=minimumDistanceIndexes[i]
                if train_data[j][0]==test_data[i][0]:
                        counter+=1
        accuracies.append(counter)
print(accuracies)
print reduce(lambda x, y: x + y, accuracies)/len(accuracies)
```

3. A) If we assume that two sets data points in a two different planes. . But, the positive points are making a circle and negative data points are also making a circle. And the positive circle and the negative circle are parallel to each other (3-D), and the distance between the two circles is slightly less than the distances between each points to others of the same circle. Thus 1-NN will always have 0% accuracies as it will get the points from opposite circle. And if we take 3-NN, a point will get two same label's data in the same circle and one opposite label from other circle. The data graph will be like the following figure:

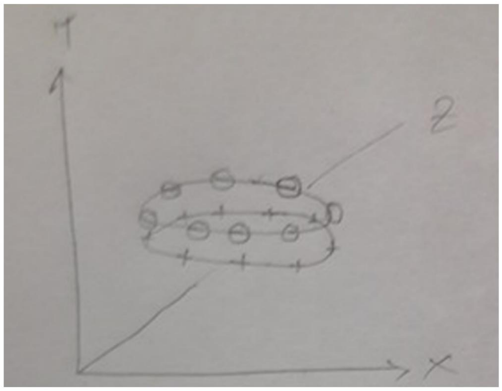


2 Question 2 40 / 40

- ✓ + **15 pts** (a) Correct
- ✓ + **15 pts** (b) Correct
- ✓ + **10 pts** (c) Correct/Adequate
- + **0 pts** (a) Accuracy values not included.
- + **5 pts** (a) Accuracy should have been decreasing.
- + **10 pts** (b) The non uniformity of the x axis scale of the graph obscures the expected graph shape.
- + **5 pts** (b) Accuracy should have been decreasing.
- + **0 pts** (b) Graph not included.
- + **0 pts** (c) Code not included.

```
        minimumDistanceIndexes=[]
        minimumDistanceIndexes=getminimumDistanceIndexes(distances)
        counter=0
        i=0
        j=0
        for i in range(len(minimumDistanceIndexes)):
            j=minimumDistanceIndexes[i]
            if train_data[j][0]==test_data[i][0]:
                counter+=1
        accuracies.append(counter)
    print(accuracies)
    print reduce(lambda x, y: x + y, accuracies)/len(accuracies)
```

3. A) If we assume that two sets data points in a two different planes. . But, the positive points are making a circle and negative data points are also making a circle. And the positive circle and the negative circle are parallel to each other (3-D), and the distance between the two circles is slightly less than the distances between each points to others of the same circle. Thus 1-NN will always have 0% accuracies as it will get the points from opposite circle. And if we take 3-NN, a point will get two same label's data in the same circle and one opposite label from other circle. The data graph will be like the following figure:



3.1 Part a 15 / 15

✓ **+ 15 pts** **Correct**

+ 13 pts Your structure is correct. But for the dataset we expected you to declare data points.

+ 10 pts You have mixed up the two distances. The height of the box should be smaller than the width and length.

+ 5 pts This data set will not give the expected result.

+ 5 pts In your case there is no way to get a 100% for 3NN in the case of - (negative) points.

+ 2 pts Your observation is correct. But we asked for you to come up with a data set.

+ 0 pts Assume a plane and put some points belonging to one class on it. Then suppose a copy of the original plane with the exact same points but with different class label. If the distance of two plains is smaller than the shortest distance of points in one plane, then we found the solution.

+ 0 pts No labels given, thus the answer is unacceptable.

+ 3 pts The question asks for the points themselves not the distances.

+ 2 pts 3-nn of point 2,2 is (-,++) which classifies it incorrectly.

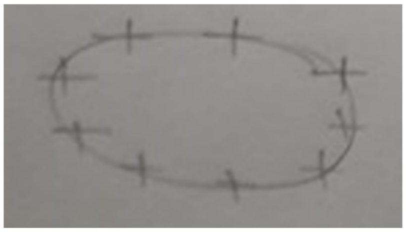
+ 3 pts You need at least 6 points.

+ 2 pts The first point is classified incorrectly, and you cannot have infinite data.

+ 0 pts Click here to replace this description.

+ 10 pts Click here to replace this description.

B) Lets assume a circle of data points with same label, which has more than 10 points. (n-1)NN will always achieve 100% accuracies.



4. For binary perceptron, every time when the model predicts wrong label, it decreases the weight by x and reduce the bias by 1.

   In multiclass perceptron, whenever the model predicts a wrong label, it decreases the weight of that predicated label and increases the weight of the actual label.

   For binary perceptron and multiclass when class=2, the models are pushing the system towards the correct class label and moving away from the wrong one. Thus, both the method will always predict the same label if the dataset are same.

   Lets have an example training dataset:

A1	A2	Label
1	0	0
1	1	0
0	1	1

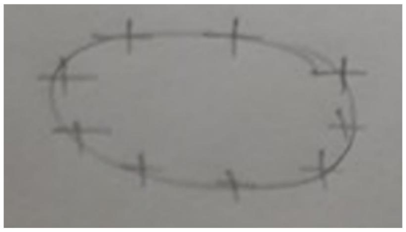
   For multiclass, let, class c1=1, c0=0

   For binary perceptron,

3.2 Part b 15 / 15

- ✓ + **15 pts** Correct
- + **2 pts** Not answered.
- + **0 pts** The solution is a dataset in which all the points have similar labels.
- + **0 pts** Click here to replace this description.
- + **2 pts** incorrect answer.

B) Lets assume a circle of data points with same label, which has more than 10 points. (n-1)NN will always achieve 100% accuracies.



4. For binary perceptron, every time when the model predicts wrong label, it decreases the weight by x and reduce the bias by 1.

   In multiclass perceptron, whenever the model predicts a wrong label, it decreases the weight of that predicated label and increases the weight of the actual label.

   For binary perceptron and multiclass when class=2, the models are pushing the system towards the correct class label and moving away from the wrong one. Thus, both the method will always predict the same label if the dataset are same.

   Lets have an example training dataset:

A1	A2	Label
1	0	0
1	1	0
0	1	1

   For multiclass, let, class c1=1, c0=0

   For binary perceptron,

Lets w1=1,w2=1,b=0

For multiclass, lets c1={w1=1, w2=1}, c2={w1=1, w2=1}, b=0

For the first row:
In binary perceptron:

$Y=w^T x+b=$ 1x1+1x0+0=1-> f(1)=1/(1+e^{-1})=0.731=~1 which is an incorrect prediction,
So, w1 and w2 will be updated, w1=1-1=0, w2=1-0=1, b=0-1=-1
So, now w1=0, w2=1, b=-1

$Y=w^T x+b=$0X1+1x0+(-1)=-1 -> f(-1)=0.2689=~0 (Correct Label)
Lets see the multiclass perceptron for the first row,

C1=1x1+1x0+0=1, c2= 1x1+1x0+0=1, both are same, so the model will pick anyone randomly. Lets it picked c1. Which is a wrong label.

Hence, it will update the weights for c1 and c2 and b.
C1={w1=1-1=0, w2=1-0=1} C2={w1=1+1=2, w2=1+0=1}, b=0+1=1

After updating the weights,
Lets see
C1=0x1+1x0-1=-1
C2=2x1+1x0+1=3

So, now it will pick C2 which is a correct label.

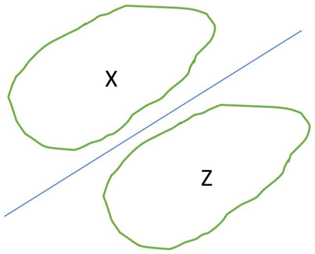
Thus, for same dataset, if the class number is two in multiclass perceptron, both the models will always predict the same label.

4 Question 4 10 / 30

✓ **- 20 pts Substantial issues: Clarity, detail, correctness**

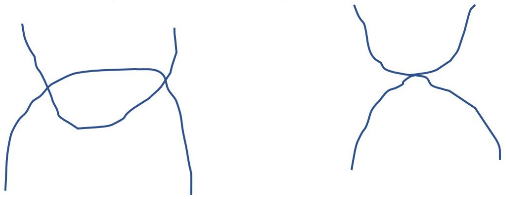
Perceptron predictions are +1/-1, not continuous as in logistic regression. Initial weights should all be zero. Perceptron update occurs when prediction is incorrect or there's a tie.

5.



$Y(x)=w^T xi+b >0$ for all xi…………(1)
$Z(z)=w^T zi + b <0$ for all zi……….(2)

Lets assume by contradiction, the two convex hulls are intersecting in a point or two:



For the intersected point:
$Y(k)= w^T xi+b= w^T zi + b$…………….(3)

Here in this point, $w^T xi+b= w^T zi + b$. But if (1) and (2) are true, (3) can not be happened as they are fully distinct: $y(x)>0$ and $y(z)<0$.

Thus it can be said that, if two convex hull intersects with each other, the data points can not be linearly separable.

5 Question 5 **30 / 30**

- ✓ **+ 30 pts** **Correct/Adequate**
- **+ 30 pts** Undergrad submission. Question N/A
- **+ 20 pts** Inadequate
- **+ 10 pts** Bonus: Both sides proved.