Project 2: Multiclass and Linear Models

By: Andrean Priadi 119407110

WU1

In [ ]:
```
from sklearn.tree import DecisionTreeClassifier
import multiclass
import util
from datasets import *
import time
```

WU1 A OAA

In [ ]:
```
h = multiclass.OAA(5, lambda: DecisionTreeClassifier(max_depth=3))
h.train(WineDataSmall.X, WineDataSmall.Y)
```

```
training classifier for 0 versus rest
training classifier for 1 versus rest
training classifier for 2 versus rest
training classifier for 3 versus rest
training classifier for 4 versus rest
```

In [ ]:
```
print(WineDataSmall.labels[0])
util.showTree(h.f[0], WineDataSmall.words)
```

```
Sauvignon-Blanc
citrus?
-N-> lime?
|     -N-> gooseberry?
|     |     -N-> class 0  (356 for class 0, 10 for class 1)
|     |     -Y-> class 1  (0 for class 0, 4 for class 1)
|     -Y-> or?
|     |     -N-> class 1  (1 for class 0, 15 for class 1)
|     |     -Y-> class 0  (2 for class 0, 0 for class 1)
-Y-> grapefruit?
|     -N-> flavors?
|     |     -N-> class 1  (4 for class 0, 12 for class 1)
|     |     -Y-> class 0  (11 for class 0, 5 for class 1)
|     -Y-> opens?
|     |     -N-> class 1  (0 for class 0, 14 for class 1)
|     |     -Y-> class 0  (1 for class 0, 0 for class 1)
```

Answer:

The most indicative words of being Sauvignon-Blanc are(not citrus and lime) The most indicative words of NOT being Sauvignon-Blanc are (not citrus and not lime and not goosebery)

In [ ]:
```
print(WineDataSmall.labels[2])
util.showTree(h.f[2], WineDataSmall.words)
```

```
Pinot-Noir
cherry?
-N-> raspberries?
|    -N-> strawberry?
|    |    -N-> class 0   (225 for class 0, 58 for class 1)
|    |    -Y-> class 1   (0 for class 0, 4 for class 1)
|    -Y-> cocoa?
|    |    -N-> class 1   (0 for class 0, 12 for class 1)
|    |    -Y-> class 0   (1 for class 0, 0 for class 1)
-Y-> cassis?
|    -N-> petit?
|    |    -N-> class 1   (36 for class 0, 68 for class 1)
|    |    -Y-> class 0   (8 for class 0, 0 for class 1)
|    -Y-> allspice?
|    |    -N-> class 0   (21 for class 0, 0 for class 1)
|    |    -Y-> class 1   (0 for class 0, 2 for class 1)
```

Answer:

The most indicative words of being Pinot-Noir are (cherry, not cassis, and not petit)
The most indicative words of NOT being Pinot-Noir are (not cherry, not raspberries, and not strawberry)

WU1 A AVA

In [ ]:
```
h = multiclass.AVA(5, lambda: DecisionTreeClassifier(max_depth=3))
h.train(WineDataSmall.X, WineDataSmall.Y)
```

```
training classifier for 1 versus 0
training classifier for 2 versus 0
training classifier for 2 versus 1
training classifier for 3 versus 0
training classifier for 3 versus 1
training classifier for 3 versus 2
training classifier for 4 versus 0
training classifier for 4 versus 1
training classifier for 4 versus 2
training classifier for 4 versus 3
```

In [ ]:
```
print("training classifier for 1 versus 0")
util.showTree(h.f[1][0], WineDataSmall.words)
```

```
training classifier for 1 versus 0
citrus?
-N-> lime?
|    -N-> refreshing?
|    |    -N-> class 0   (187 for class 0, 9 for class 1)
|    |    -Y-> class 1   (0 for class 0, 5 for class 1)
|    -Y-> class 1        (0 for class 0, 15 for class 1)
-Y-> class 1    (0 for class 0, 31 for class 1)
```

In [ ]:
```
print("training classifier for 2 versus 0")
util.showTree(h.f[2][0], WineDataSmall.words)
```

```
training classifier for 2 versus 0
crisp?
-N-> lime?
|    -N-> lemon?
|    |    -N-> class 0  (141 for class 0, 9 for class 1)
|    |    -Y-> class 1  (0 for class 0, 8 for class 1)
|    -Y-> persistence?
|    |    -N-> class 1  (0 for class 0, 13 for class 1)
|    |    -Y-> class 0  (1 for class 0, 0 for class 1)
-Y-> red?
|    -N-> class 1       (0 for class 0, 30 for class 1)
|    -Y-> class 0       (2 for class 0, 0 for class 1)
```

In [ ]:
```
print("training classifier for 3 versus 0")
util.showTree(h.f[3][0], WineDataSmall.words)
```

```
training classifier for 3 versus 0
thai?
-N-> very?
|    -N-> between?
|    |    -N-> class 1  (4 for class 0, 56 for class 1)
|    |    -Y-> class 0  (1 for class 0, 0 for class 1)
|    -Y-> ripe?
|    |    -N-> class 1  (1 for class 0, 4 for class 1)
|    |    -Y-> class 0  (4 for class 0, 0 for class 1)
-Y-> class 0    (5 for class 0, 0 for class 1)
```

In [ ]:
```
print("training classifier for 4 versus 0")
util.showTree(h.f[4][0], WineDataSmall.words)
```

```
training classifier for 4 versus 0
apple?
-N-> pasta?
|    -N-> quite?
|    |    -N-> class 1  (11 for class 0, 56 for class 1)
|    |    -Y-> class 0  (3 for class 0, 0 for class 1)
|    -Y-> class 0       (4 for class 0, 0 for class 1)
-Y-> bright?
|    -N-> class 0       (10 for class 0, 0 for class 1)
|    -Y-> particularly?
|    |    -N-> class 1  (0 for class 0, 4 for class 1)
|    |    -Y-> class 0  (1 for class 0, 0 for class 1)
```

Answer:

The most indicative words of being Sauvignon-Blanc are lime, (crisp and not red), (not thai, not very, not been), (not apple, not pasta, not quite) The most indicative words of NOT being Sauvignon-Blanc are (not lime, not citrus, not refreshing), not lemon

In [ ]:
```
print("training classifier for 2 versus 0")
util.showTree(h.f[2][0], WineDataSmall.words)
```

```
training classifier for 2 versus 0
crisp?
-N-> lime?
|    -N-> lemon?
|    |    -N-> class 0  (141 for class 0, 9 for class 1)
|    |    -Y-> class 1  (0 for class 0, 8 for class 1)
|    -Y-> persistence?
|    |    -N-> class 1  (0 for class 0, 13 for class 1)
|    |    -Y-> class 0  (1 for class 0, 0 for class 1)
-Y-> red?
|    -N-> class 1       (0 for class 0, 30 for class 1)
|    -Y-> class 0       (2 for class 0, 0 for class 1)
```

In [ ]:
```
print("training classifier for 2 versus 1")
util.showTree(h.f[2][1], WineDataSmall.words)
```

```
training classifier for 2 versus 1
cassis?
-N-> acidity?
|    -N-> salmon?
|    |    -N-> class 1  (92 for class 0, 129 for class 1)
|    |    -Y-> class 0  (11 for class 0, 0 for class 1)
|    -Y-> tannins?
|    |    -N-> class 0  (22 for class 0, 0 for class 1)
|    |    -Y-> class 0  (15 for class 0, 11 for class 1)
-Y-> tea?
|    -N-> 100?
|    |    -N-> class 1  (1 for class 0, 47 for class 1)
|    |    -Y-> class 0  (1 for class 0, 0 for class 1)
|    -Y-> class 0       (2 for class 0, 0 for class 1)
```

In [ ]:
```
print("training classifier for 3 versus 2")
util.showTree(h.f[3][2], WineDataSmall.words)
```

```
training classifier for 3 versus 2
crisp?
-N-> peach?
|    -N-> pear?
|    |    -N-> class 1  (3 for class 0, 142 for class 1)
|    |    -Y-> class 0  (2 for class 0, 0 for class 1)
|    -Y-> class 0       (3 for class 0, 0 for class 1)
-Y-> red?
|    -N-> class 0       (7 for class 0, 0 for class 1)
|    -Y-> class 1       (0 for class 0, 2 for class 1)
```

In [ ]:
```
print("training classifier for 4 versus 2")
util.showTree(h.f[4][2], WineDataSmall.words)
```

```
training classifier for 4 versus 2
straw?
-N-> crisp?
|    -N-> example?
|    |    -N-> class 1  (8 for class 0, 142 for class 1)
|    |    -Y-> class 0  (2 for class 0, 0 for class 1)
|    -Y-> red?
|    |    -N-> class 0  (7 for class 0, 0 for class 1)
|    |    -Y-> class 1  (0 for class 0, 2 for class 1)
-Y-> class 0    (12 for class 0, 0 for class 1)
```

Answer:

The most indicative words of being Pinot-Noir are cassis, straw The most indicative words of NOT being Pinot-Noir are crisp

WU1 B

Train depth 3 decision trees on the full WineData task (With 20 labels). What accuracy do you get? How long does this take (in seconds)? One of my least favorite wine is Viognier -- what words are indicative of this?

WU1 B OAA

In [ ]:
```
start = time.time()
h = multiclass.OAA(20, lambda: DecisionTreeClassifier(max_depth=3))
end = time.time()
h.train(WineData.X, WineData.Y)
P = h.predictAll(WineData.Xte)
acc = mean(P == WineData.Yte)
```

```
training classifier for 0 versus rest
training classifier for 1 versus rest
training classifier for 2 versus rest
training classifier for 3 versus rest
training classifier for 4 versus rest
training classifier for 5 versus rest
training classifier for 6 versus rest
training classifier for 7 versus rest
training classifier for 8 versus rest
training classifier for 9 versus rest
training classifier for 10 versus rest
training classifier for 11 versus rest
training classifier for 12 versus rest
training classifier for 13 versus rest
training classifier for 14 versus rest
training classifier for 15 versus rest
training classifier for 16 versus rest
training classifier for 17 versus rest
training classifier for 18 versus rest
training classifier for 19 versus rest
```

In [ ]:
```
print ('OAA accuracy: %f' % acc)
print ("OAA training time taken:", end - start, "seconds")
```

```
OAA accuracy: 0.368275
OAA training time taken: 0.00016808509826660156 seconds
```

In [ ]:
```
print(WineData.labels[17])
util.showTree(h.f[17], WineData.words)
```

```
Viognier
peaches?
-N-> nectarine?
|    -N-> chilled?
|    |    -N-> class 0  (1036 for class 0, 1 for class 1)
|    |    -Y-> class 0  (6 for class 0, 1 for class 1)
|    -Y-> savory?
|    |    -N-> class 0  (13 for class 0, 1 for class 1)
|    |    -Y-> class 1  (0 for class 0, 1 for class 1)
-Y-> milk?
|    -N-> brilliant?
|    |    -N-> class 0  (14 for class 0, 0 for class 1)
|    |    -Y-> class 1  (0 for class 0, 1 for class 1)
|    -Y-> class 1       (0 for class 0, 3 for class 1)
```

Most indicative words for Viognier are milk

Most not indiciate words for Viognier are peaches

In [ ]:
```
start = time.time()
h = multiclass.AVA(20, lambda: DecisionTreeClassifier(max_depth=1))
h.train(WineData.X, WineData.Y)
end = time.time()
P = h.predictAll(WineData.Xte)
acc = mean(P == WineData.Yte)
```

```
training classifier for 1 versus 0
```

```
training classifier for 2 versus 0
training classifier for 2 versus 1
training classifier for 3 versus 0
training classifier for 3 versus 1
training classifier for 3 versus 2
training classifier for 4 versus 0
training classifier for 4 versus 1
training classifier for 4 versus 2
training classifier for 4 versus 3
training classifier for 5 versus 0
training classifier for 5 versus 1
training classifier for 5 versus 2
training classifier for 5 versus 3
training classifier for 5 versus 4
training classifier for 6 versus 0
training classifier for 6 versus 1
training classifier for 6 versus 2
training classifier for 6 versus 3
training classifier for 6 versus 4
training classifier for 6 versus 5
training classifier for 7 versus 0
training classifier for 7 versus 1
training classifier for 7 versus 2
training classifier for 7 versus 3
training classifier for 7 versus 4
training classifier for 7 versus 5
training classifier for 7 versus 6
training classifier for 8 versus 0
training classifier for 8 versus 1
training classifier for 8 versus 2
training classifier for 8 versus 3
training classifier for 8 versus 4
training classifier for 8 versus 5
training classifier for 8 versus 6
training classifier for 8 versus 7
training classifier for 9 versus 0
training classifier for 9 versus 1
training classifier for 9 versus 2
training classifier for 9 versus 3
training classifier for 9 versus 4
training classifier for 9 versus 5
training classifier for 9 versus 6
training classifier for 9 versus 7
training classifier for 9 versus 8
training classifier for 10 versus 0
training classifier for 10 versus 1
training classifier for 10 versus 2
training classifier for 10 versus 3
training classifier for 10 versus 4
training classifier for 10 versus 5
training classifier for 10 versus 6
training classifier for 10 versus 7
training classifier for 10 versus 8
training classifier for 10 versus 9
training classifier for 11 versus 0
training classifier for 11 versus 1
training classifier for 11 versus 2
```

```
training classifier for 11 versus 3
training classifier for 11 versus 4
training classifier for 11 versus 5
training classifier for 11 versus 6
training classifier for 11 versus 7
training classifier for 11 versus 8
training classifier for 11 versus 9
training classifier for 11 versus 10
training classifier for 12 versus 0
training classifier for 12 versus 1
training classifier for 12 versus 2
training classifier for 12 versus 3
training classifier for 12 versus 4
training classifier for 12 versus 5
training classifier for 12 versus 6
training classifier for 12 versus 7
training classifier for 12 versus 8
training classifier for 12 versus 9
training classifier for 12 versus 10
training classifier for 12 versus 11
training classifier for 13 versus 0
training classifier for 13 versus 1
training classifier for 13 versus 2
training classifier for 13 versus 3
training classifier for 13 versus 4
training classifier for 13 versus 5
training classifier for 13 versus 6
training classifier for 13 versus 7
training classifier for 13 versus 8
training classifier for 13 versus 9
training classifier for 13 versus 10
training classifier for 13 versus 11
training classifier for 13 versus 12
training classifier for 14 versus 0
training classifier for 14 versus 1
training classifier for 14 versus 2
training classifier for 14 versus 3
training classifier for 14 versus 4
training classifier for 14 versus 5
training classifier for 14 versus 6
training classifier for 14 versus 7
training classifier for 14 versus 8
training classifier for 14 versus 9
training classifier for 14 versus 10
training classifier for 14 versus 11
training classifier for 14 versus 12
training classifier for 14 versus 13
training classifier for 15 versus 0
training classifier for 15 versus 1
training classifier for 15 versus 2
training classifier for 15 versus 3
training classifier for 15 versus 4
training classifier for 15 versus 5
training classifier for 15 versus 6
training classifier for 15 versus 7
training classifier for 15 versus 8
training classifier for 15 versus 9
```

```
training classifier for 15 versus 10
training classifier for 15 versus 11
training classifier for 15 versus 12
training classifier for 15 versus 13
training classifier for 15 versus 14
training classifier for 16 versus 0
training classifier for 16 versus 1
training classifier for 16 versus 2
training classifier for 16 versus 3
training classifier for 16 versus 4
training classifier for 16 versus 5
training classifier for 16 versus 6
training classifier for 16 versus 7
training classifier for 16 versus 8
training classifier for 16 versus 9
training classifier for 16 versus 10
training classifier for 16 versus 11
training classifier for 16 versus 12
training classifier for 16 versus 13
training classifier for 16 versus 14
training classifier for 16 versus 15
training classifier for 17 versus 0
training classifier for 17 versus 1
training classifier for 17 versus 2
training classifier for 17 versus 3
training classifier for 17 versus 4
training classifier for 17 versus 5
training classifier for 17 versus 6
training classifier for 17 versus 7
training classifier for 17 versus 8
training classifier for 17 versus 9
training classifier for 17 versus 10
training classifier for 17 versus 11
training classifier for 17 versus 12
training classifier for 17 versus 13
training classifier for 17 versus 14
training classifier for 17 versus 15
training classifier for 17 versus 16
training classifier for 18 versus 0
training classifier for 18 versus 1
training classifier for 18 versus 2
training classifier for 18 versus 3
training classifier for 18 versus 4
training classifier for 18 versus 5
training classifier for 18 versus 6
training classifier for 18 versus 7
training classifier for 18 versus 8
training classifier for 18 versus 9
training classifier for 18 versus 10
training classifier for 18 versus 11
training classifier for 18 versus 12
training classifier for 18 versus 13
training classifier for 18 versus 14
training classifier for 18 versus 15
training classifier for 18 versus 16
training classifier for 18 versus 17
training classifier for 19 versus 0
```

```
training classifier for 19 versus 1
training classifier for 19 versus 2
training classifier for 19 versus 3
training classifier for 19 versus 4
training classifier for 19 versus 5
training classifier for 19 versus 6
training classifier for 19 versus 7
training classifier for 19 versus 8
training classifier for 19 versus 9
training classifier for 19 versus 10
training classifier for 19 versus 11
training classifier for 19 versus 12
training classifier for 19 versus 13
training classifier for 19 versus 14
training classifier for 19 versus 15
training classifier for 19 versus 16
training classifier for 19 versus 17
training classifier for 19 versus 18
```

In [ ]:
```
print ('AVA accuracy: %f' % acc)
print ("AVA training time taken:", end - start, "seconds")
```

```
AVA accuracy: 0.212430
AVA training time taken: 0.24743103981018066 seconds
```

WU1 C

WU1 C OAA

In [ ]:
```
h = multiclass.OAA(20, lambda: DecisionTreeClassifier(max_depth=3))
h.train(WineData.X, WineData.Y)
```

```
training classifier for 0 versus rest
training classifier for 1 versus rest
training classifier for 2 versus rest
training classifier for 3 versus rest
training classifier for 4 versus rest
training classifier for 5 versus rest
training classifier for 6 versus rest
training classifier for 7 versus rest
training classifier for 8 versus rest
training classifier for 9 versus rest
training classifier for 10 versus rest
training classifier for 11 versus rest
training classifier for 12 versus rest
training classifier for 13 versus rest
training classifier for 14 versus rest
training classifier for 15 versus rest
training classifier for 16 versus rest
training classifier for 17 versus rest
training classifier for 18 versus rest
training classifier for 19 versus rest
```

In [ ]:
```
P = h.predictAll(WineData.Xte, useZeroOne=False)
mean(P == WineData.Yte)
```

Out[ ]:
```
0.37105751391465674
```

In [ ]:
```
P = h.predictAll(WineData.Xte, useZeroOne=True)
mean(P == WineData.Yte)
```

Out[ ]:
```
0.24953617810760667
```

AVA

In [ ]:
```
h = multiclass.AVA(20, lambda: DecisionTreeClassifier(max_depth=3))
h.train(WineData.X, WineData.Y)
```

```
training classifier for 1 versus 0
training classifier for 2 versus 0
training classifier for 2 versus 1
training classifier for 3 versus 0
training classifier for 3 versus 1
training classifier for 3 versus 2
training classifier for 4 versus 0
training classifier for 4 versus 1
training classifier for 4 versus 2
training classifier for 4 versus 3
training classifier for 5 versus 0
training classifier for 5 versus 1
training classifier for 5 versus 2
training classifier for 5 versus 3
training classifier for 5 versus 4
training classifier for 6 versus 0
training classifier for 6 versus 1
training classifier for 6 versus 2
training classifier for 6 versus 3
training classifier for 6 versus 4
training classifier for 6 versus 5
training classifier for 7 versus 0
training classifier for 7 versus 1
training classifier for 7 versus 2
training classifier for 7 versus 3
training classifier for 7 versus 4
training classifier for 7 versus 5
training classifier for 7 versus 6
training classifier for 8 versus 0
training classifier for 8 versus 1
training classifier for 8 versus 2
training classifier for 8 versus 3
training classifier for 8 versus 4
training classifier for 8 versus 5
training classifier for 8 versus 6
training classifier for 8 versus 7
training classifier for 9 versus 0
training classifier for 9 versus 1
training classifier for 9 versus 2
training classifier for 9 versus 3
training classifier for 9 versus 4
training classifier for 9 versus 5
training classifier for 9 versus 6
```

```
training classifier for 9 versus 7
training classifier for 9 versus 8
training classifier for 10 versus 0
training classifier for 10 versus 1
training classifier for 10 versus 2
training classifier for 10 versus 3
training classifier for 10 versus 4
training classifier for 10 versus 5
training classifier for 10 versus 6
training classifier for 10 versus 7
training classifier for 10 versus 8
training classifier for 10 versus 9
training classifier for 11 versus 0
training classifier for 11 versus 1
training classifier for 11 versus 2
training classifier for 11 versus 3
training classifier for 11 versus 4
training classifier for 11 versus 5
training classifier for 11 versus 6
training classifier for 11 versus 7
training classifier for 11 versus 8
training classifier for 11 versus 9
training classifier for 11 versus 10
training classifier for 12 versus 0
training classifier for 12 versus 1
training classifier for 12 versus 2
training classifier for 12 versus 3
training classifier for 12 versus 4
training classifier for 12 versus 5
training classifier for 12 versus 6
training classifier for 12 versus 7
training classifier for 12 versus 8
training classifier for 12 versus 9
training classifier for 12 versus 10
training classifier for 12 versus 11
training classifier for 13 versus 0
training classifier for 13 versus 1
training classifier for 13 versus 2
training classifier for 13 versus 3
training classifier for 13 versus 4
training classifier for 13 versus 5
training classifier for 13 versus 6
training classifier for 13 versus 7
training classifier for 13 versus 8
training classifier for 13 versus 9
training classifier for 13 versus 10
training classifier for 13 versus 11
training classifier for 13 versus 12
training classifier for 14 versus 0
training classifier for 14 versus 1
training classifier for 14 versus 2
training classifier for 14 versus 3
training classifier for 14 versus 4
training classifier for 14 versus 5
training classifier for 14 versus 6
training classifier for 14 versus 7
training classifier for 14 versus 8
```

```
training classifier for 14 versus 9
training classifier for 14 versus 10
training classifier for 14 versus 11
training classifier for 14 versus 12
training classifier for 14 versus 13
training classifier for 15 versus 0
training classifier for 15 versus 1
training classifier for 15 versus 2
training classifier for 15 versus 3
training classifier for 15 versus 4
training classifier for 15 versus 5
training classifier for 15 versus 6
training classifier for 15 versus 7
training classifier for 15 versus 8
training classifier for 15 versus 9
training classifier for 15 versus 10
training classifier for 15 versus 11
training classifier for 15 versus 12
training classifier for 15 versus 13
training classifier for 15 versus 14
training classifier for 16 versus 0
training classifier for 16 versus 1
training classifier for 16 versus 2
training classifier for 16 versus 3
training classifier for 16 versus 4
training classifier for 16 versus 5
training classifier for 16 versus 6
training classifier for 16 versus 7
training classifier for 16 versus 8
training classifier for 16 versus 9
training classifier for 16 versus 10
training classifier for 16 versus 11
training classifier for 16 versus 12
training classifier for 16 versus 13
training classifier for 16 versus 14
training classifier for 16 versus 15
training classifier for 17 versus 0
training classifier for 17 versus 1
training classifier for 17 versus 2
training classifier for 17 versus 3
training classifier for 17 versus 4
training classifier for 17 versus 5
training classifier for 17 versus 6
training classifier for 17 versus 7
training classifier for 17 versus 8
training classifier for 17 versus 9
training classifier for 17 versus 10
training classifier for 17 versus 11
training classifier for 17 versus 12
training classifier for 17 versus 13
training classifier for 17 versus 14
training classifier for 17 versus 15
training classifier for 17 versus 16
training classifier for 18 versus 0
training classifier for 18 versus 1
training classifier for 18 versus 2
training classifier for 18 versus 3
```

```
training classifier for 18 versus 4
training classifier for 18 versus 5
training classifier for 18 versus 6
training classifier for 18 versus 7
training classifier for 18 versus 8
training classifier for 18 versus 9
training classifier for 18 versus 10
training classifier for 18 versus 11
training classifier for 18 versus 12
training classifier for 18 versus 13
training classifier for 18 versus 14
training classifier for 18 versus 15
training classifier for 18 versus 16
training classifier for 18 versus 17
training classifier for 19 versus 0
training classifier for 19 versus 1
training classifier for 19 versus 2
training classifier for 19 versus 3
training classifier for 19 versus 4
training classifier for 19 versus 5
training classifier for 19 versus 6
training classifier for 19 versus 7
training classifier for 19 versus 8
training classifier for 19 versus 9
training classifier for 19 versus 10
training classifier for 19 versus 11
training classifier for 19 versus 12
training classifier for 19 versus 13
training classifier for 19 versus 14
training classifier for 19 versus 15
training classifier for 19 versus 16
training classifier for 19 versus 17
training classifier for 19 versus 18
```

In [ ]:
```
P = h.predictAll(WineData.Xte, useZeroOne=False)
mean(P == WineData.Yte)
```

Out[ ]:
```
0.2653061224489796
```

In [ ]:
```
P = h.predictAll(WineData.Xte, useZeroOne=True)
mean(P == WineData.Yte)
```

Out[ ]:
```
0.2634508348794063
```

WU2

In [ ]:
```
t = multiclass.makeBalancedTree(range(20))
h = multiclass.MCTree(t, lambda: DecisionTreeClassifier(max_depth=3))
h.train(WineData.X, WineData.Y)
P = h.predictAll(WineData.Xte)
mean(P == WineData.Yte)
```

```
training classifier for [0, 1, 2, 3, 4, 5, 6, 7, 8, 9] versus [10, 11, 12
, 13, 14, 15, 16, 17, 18, 19]
training classifier for [0, 1, 2, 3, 4] versus [5, 6, 7, 8, 9]
training classifier for [0, 1] versus [2, 3, 4]
training classifier for [0] versus [1]
training classifier for [2] versus [3, 4]
training classifier for [3] versus [4]
training classifier for [5, 6] versus [7, 8, 9]
training classifier for [5] versus [6]
training classifier for [7] versus [8, 9]
training classifier for [8] versus [9]
training classifier for [10, 11, 12, 13, 14] versus [15, 16, 17, 18, 19]
training classifier for [10, 11] versus [12, 13, 14]
training classifier for [10] versus [11]
training classifier for [12] versus [13, 14]
training classifier for [13] versus [14]
training classifier for [15, 16] versus [17, 18, 19]
training classifier for [15] versus [16]
training classifier for [17] versus [18, 19]
training classifier for [18] versus [19]
```

Out[ ]:
```
0.3098330241187384
```

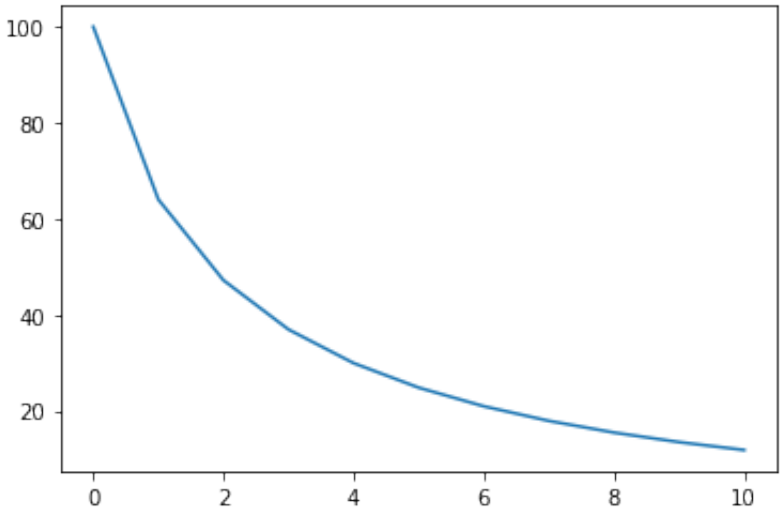
WU3

In [ ]:
```
import gd
import matplotlib.pyplot as plt
```

In [ ]:
```
x, trajectory = gd.gd(lambda x: x**2, lambda x: 2*x, 10, 10, 0.1)
```

In [ ]:
```
plt.plot(trajectory)
```

Out[ ]:
```
[<matplotlib.lines.Line2D at 0x1269a4d90>]
```

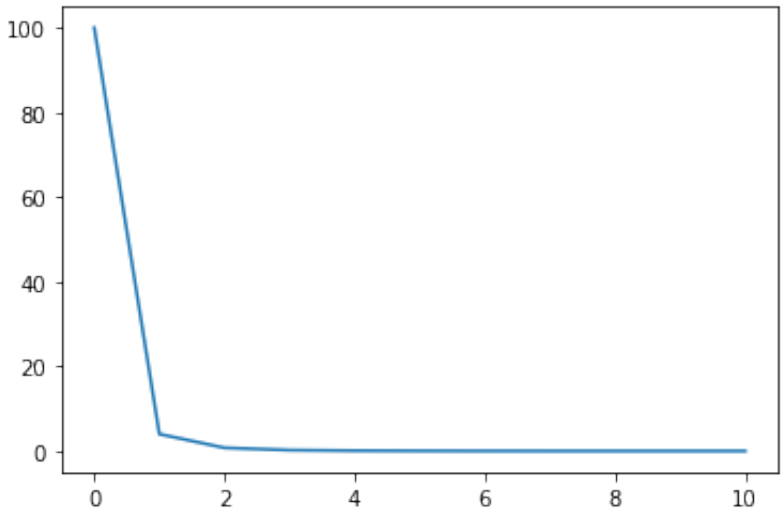


In [ ]:
```
x, trajectory = gd.gd(lambda x: x**2, lambda x: 2*x, 10, 10, 0.4)
x
```

Out[ ]:
```
0.033242949530843946
```

In [ ]:
```
plt.plot(trajectory)
```

Out[ ]:
```
[<matplotlib.lines.Line2D at 0x126a649d0>]
```



WU3 Answer:

Step size plays an important role for us to find the local minima. It should not be too small or too large. Hence, it should not too diverging or too converging

From above example:

when step size = 0.1, it diverges.

when step size = 0.4, it converges.

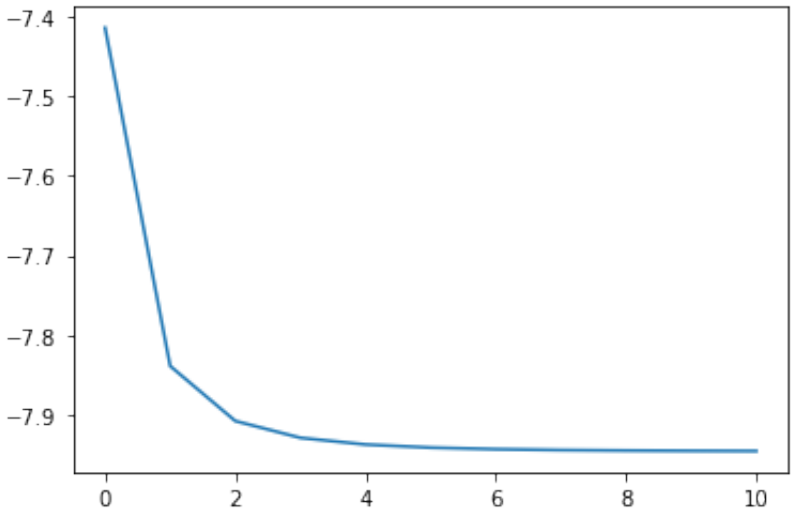
WU4

In [ ]:
```
from math import *
# non convex function (Global minimum)
x, trajectory = gd.gd(lambda x: x**2 + 10*sin(x), lambda x: 2*x + 10*cos(
x
```

Out[ ]:
```
-1.2975801408641767
```

In [ ]:
```
plt.plot(trajectory)
```

Out[ ]:
```
[<matplotlib.lines.Line2D at 0x126aa9e10>]
```

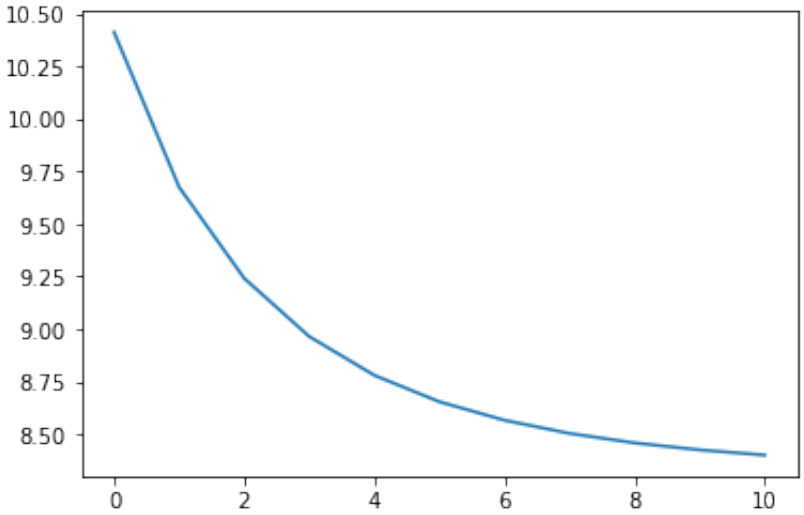


In [ ]:
```
from math import *
# non convex function (local minimum)
x, trajectory = gd.gd(lambda x: x**2 + 10*sin(x), lambda x: 2*x + 10*cos(
x
```

Out[ ]:
```
3.6919703328267657
```

In [ ]:
```
plt.plot(trajectory)
```

Out[ ]:
```
[<matplotlib.lines.Line2D at 0x126af2bf0>]
```



WU5

In [ ]:
```
import runClassifier
import linear
import datasets
import mlGraphics
```

In [ ]:
```
f = linear.LinearClassifier({'lossFunction': linear.SquaredLoss(), 'lambd
runClassifier.trainTestSet(f, datasets.WineDataBinary)
```

```
Training accuracy 0.242915, test accuracy 0.313653
```

In [ ]:
```
f = linear.LinearClassifier({'lossFunction': linear.LogisticLoss(), 'lamb
runClassifier.trainTestSet(f, datasets.WineDataBinary)
```

```
Training accuracy 0.995951, test accuracy 0.97417
```

In [ ]:
```
arr = list(f.getRepresentation())
data = []
for i in range(len(datasets.WineDataBinary.words)):
    data.append((arr[i], datasets.WineDataBinary.words[i]))
data.sort()
data
```

Out[ ]:
```
[(-1.1695212164040434, 'tannins'),
 (-0.7653093906427076, 'black'),
 (-0.6835931677893788, 'dark'),
 (-0.6295907281434356, 'cherry'),
 (-0.5321916724675303, 'blackberry'),
 (-0.5092846891935376, 'deep'),
 (-0.4742724034212623, 'red'),
 (-0.464098396577844, 'oak'),
 (-0.4393804627551591, 'berry'),
 (-0.4306169644161318, 'chocolate'),
 (-0.4202651419017117, 'currant'),
 (-0.3974537952686579, 'plum'),
 (-0.39673551395193246, 'soft'),
 (-0.3651026008004167, 'vanilla'),
 (-0.3588029028532654, 'years'),
 (-0.34944865069051934, 'structure'),
 (-0.33879523379511506, 'cassis'),
 (-0.3359318389706689, 'flavors'),
 (-0.3274528079180455, 'cocoa'),
 (-0.3005539369064018, 'spice'),
 (-0.28375576962558524, 'valley'),
 (-0.2653345344211855, 'tobacco'),
 (-0.2640374487031105, 'is'),
 (-0.2601717767845841, 'supple'),
 (-0.2555187032348373, 'very'),
 (-0.25497141259624234, ';'),
 (-0.24962822438910076, 'lush'),
 (-0.24886970249942986, 'rich'),
 (-0.245575367141781, 'vintage'),
 (-0.24221421404409277, 'verdot'),
 (-0.23707252760696843, 'full-bodied'),
 (-0.23471611755228672, 'can'),
 (-0.23231371811502588, 'berries'),
 (-0.22795170230775483, 'youthful'),
 (-0.2270028479622386, 'cherries'),
 (-0.22591332309076378, 'be'),
 (-0.22560721341255846, 'smooth'),
 (-0.22396668670618422, 'you'),
 (-0.2216704242049923, 'licorice'),
 (-0.2209909885411165, 'medium'),
 (-0.20877786215825234, 'long'),
 (-0.20874946081167312, 'espresso'),
 (-0.20400249443017376, 'mocha'),
```

```
(-0.20223794180964197, 'silky'),
(-0.20196835066916252, 'cedar'),
(-0.20126511028417957, 'petit'),
(-0.20056409452917348, 'raspberry'),
(-0.19719674206038554, 'concentrated'),
(-0.19594021898909408, 'purple'),
(-0.19283328631090685, 'spices'),
(-0.1875020469525165, 'texture'),
(-0.18718287386202093, 'cola'),
(-0.18464525575178029, '5'),
(-0.1834218559315892, 'mint'),
(-0.18337318076768808, '20'),
(-0.17908024566539166, 'round'),
(-0.17907110483694594, 'to'),
(-0.17877676415684984, '&'),
(-0.17738856184412952, 'each'),
(-0.17334571962176507, 'elegant'),
(-0.17317719439203194, 'structured'),
(-0.17316677270083103, 'finish'),
(-0.16837262092819244, 'textured'),
(-0.16672189601165774, '2009'),
(-0.16408209349212474, 'coupled'),
(-0.16403397911050843, 'expression'),
(-0.16379707447097863, 'crimson'),
(-0.16348975549371755, 'wine'),
(-0.15906163864041228, 'will'),
(-0.1585149405973172, 'nuances'),
(-0.15809940324277036, 'roasted'),
(-0.15794908161393476, 'forward'),
(-0.1543856529557695, 'velvety'),
(-0.1536278028908627, 'lead'),
(-0.15003089723484697, 'warm'),
(-0.149870751327968, 'shows'),
(-0.1480087570390206, 'franc'),
(-0.14718252362996684, 'wines'),
(-0.1466346605826882, 'balanced'),
(-0.145983389910992, 'napa'),
(-0.14556728668306193, 'mountain'),
(-0.14327487855667373, 'new'),
(-0.14155335701504843, 'big'),
(-0.14138755858286414, 'hints'),
(-0.14084248020007173, 'dry'),
(-0.13916119180760683, 'hint'),
(-0.13704054838726884, 'firm'),
(-0.1369072920211587, 'rose'),
(-0.136572190342604, 'seamlessly'),
(-0.13528052967755083, 'sweet'),
(-0.13446791082272813, 'grilled'),
(-0.13347815909357522, 'estate'),
(-0.133245465483266, 'integrated'),
(-0.13282820779923252, '2'),
(-0.13261328491499744, 'herb'),
(-0.1307209749862375, 'at'),
(-0.12988970238669217, 'tannin'),
(-0.12973962855620444, 'plums'),
(-0.1292336627918294, 'enticing'),
(-0.1281624077648053, 'but'),
```

```
(-0.12670364897150066, 'cheeses'),
(-0.12598930901585023, 'maturity'),
(-0.12540481938858383, 'broad'),
(-0.1252078348964383, 'robust'),
(-0.12333427334516688, 'coffee'),
(-0.12202954646609623, 'violet'),
(-0.12177088835739872, 'or'),
(-0.12086068450085813, 'beautifully'),
(-0.12029766241088147, 'all'),
(-0.1192161646184038, 'on'),
(-0.11891005181431495, 'steak'),
(-0.11862555164050342, 'sugar'),
(-0.11808631170436809, 'for'),
(-0.11785892138975299, 'region'),
(-0.11784005229917949, 'dense'),
(-0.11728196203084075, 'power'),
(-0.11627046448346338, 'blueberry'),
(-0.11590759527608485, 'leap'),
(-0.1147996141723572, 'hue'),
(-0.11471105493010983, 'ruby'),
(-0.1142408695860009, 'jammy'),
(-0.11356320590268931, 'now'),
(-0.11309776071510422, 'in'),
(-0.11302428841075633, 'winemaker'),
(-0.11239472648749298, 'mouth'),
(-0.1119732820134568, 'french'),
(-0.11166370984450086, '2008'),
(-0.11098791865184589, 'beef'),
(-0.10993426246394318, 'earth'),
(-0.10956591718757062, 'nutmeg'),
(-0.10869750824021435, 'blackberries'),
(-0.1072089193343243, 'caramel'),
(-0.10692614385558304, 'pair'),
(-0.10566918390218888, 'pepper'),
(-0.10471289925804253, 'soils'),
(-0.10408185448857787, 'amp'),
(-0.10376096780555044, 'expressive'),
(-0.10252685019379726, 'more'),
(-0.10172983458084589, 'world'),
(-0.10138455050359141, 'gentle'),
(-0.09889269059374728, 'drink'),
(-0.09851669814207227, 'yet'),
(-0.09834406427190494, 'jam'),
(-0.09755785208816485, 'who'),
(-0.09747960702970909, 'concentration'),
(-0.09739763944111311, 'than'),
(-0.09720802104577315, 'bodied'),
(-0.0971239741505289, 'one'),
(-0.09681706091726386, 'has'),
(-0.09658129782508328, 'alluring'),
(-0.09610045267349601, 'over'),
(-0.0955812016920441 8, 'depth'),
(-0.0955074770224579, 'develop'),
(-0.09449890850702167, 'leather'),
(-0.0943686577862992, 'meat'),
(-0.09325182132892254, 'crafted'),
(-0.09229340968931242, 'opens'),
```

```
(-0.09208909042277603, 'nice'),
(-0.09110680309619539, 't'),
(-0.08938282905060413, 'first'),
(-0.08798039238876024, 'hearty'),
(-0.087859872880703, 'different'),
(-0.08715349811141489, 'garnet'),
(-0.08630825732824468, 'throughout'),
(-0.08551284658439899, '10'),
(-0.08543292728641103, 'background'),
(-0.08520231155984903, 'planted'),
(-0.08520231155984903, 'vines'),
(-0.08331088404771228, 'just'),
(-0.08197040501635644, 'california'),
(-0.08179818268611272, 'spicy'),
(-0.08171336841246646, 'find'),
(-0.08137408011208663, 'seductive'),
(-0.07921816783656571, 'Â'),
(-0.07809806980982521, 'time'),
(-0.07807342632382962, 'violets'),
(-0.07795690644980853, 'its'),
(-0.07770031947561, '#'),
(-0.07770031947561, '39'),
(-0.0774462328319239, 'sage'),
(-0.07689597054504652, 'year'),
(-0.07660701252394199, 'aroma'),
(-0.07525655027337501, 'reserve'),
(-0.07524445795456869, '2007'),
(-0.07514381688723876, 'days'),
(-0.0747222307272516, 'earthy'),
(-0.07402481914196735, 'de'),
(-0.07389626117999472, 'these'),
(-0.07356252912479307, 'meats'),
(-0.0728049599538394, 'plush'),
(-0.07233698973814291, 'followed'),
(-0.07206872231643519, 'bean'),
(-0.07181663333647188, 'body'),
(-0.07154198505716511, 'like'),
(-0.0700582185429427 8, 'make'),
(-0.0698563546806035, 'intriguing'),
(-0.06956569408136633, 'finishing'),
(-0.06930475222993442, 'sip'),
(-0.06903040832838143, 'vineyards'),
(-0.06899379770925265, 'medium-bodied'),
(-0.06871736722350591, 'five'),
(-0.0683726563630987, 'don'),
(-0.06790351465697085, 'young'),
(-0.06766679110275085, 'drinking'),
(-0.06745382943820277, 'pork'),
(-0.06696799389965183, 'located'),
(-0.06633087209592826, 'soil'),
(-0.065657956343167, 'rounded'),
(-0.06553718094946252, 'which'),
(-0.06531813252949445, 'other'),
(-0.0650135452134861, 'we'),
(-0.06501155266694414, 'powerful'),
(-0.06500896407615461, 'anise'),
(-0.06448527998630157, 'tea'),
```

```
(-0.06366390693586055, 'blue'),
(-0.06327002915304294, 'slight'),
(-0.06323492042034096, 'fine'),
(-0.06277274503071427, 'opulent'),
(-0.061475651947372735, 'should'),
(-0.06124173291936688, 'pasta'),
(-0.06097891890861895, 'before'),
(-0.06052866831085708, 'reveals'),
(-0.06045765667585278, 'beautiful'),
(-0.06031410167804243, 'gracefully'),
(-0.05964208945289361, '3'),
(-0.05896290882344881, 'still'),
(-0.05891258183666053, 'vine'),
(-0.057714184421229844, 'smoky'),
(-0.057507960075258484, 'varietal'),
(-0.0574863765511379, 'release'),
(-0.05744875741120424, 'lasting'),
(-0.05692790041678152, 'Ã'),
(-0.05689376346704517, 'upon'),
(-0.05660317621455602, 'smoke'),
(-0.05594851252082733, 'fill'),
(-0.05489619845087708, 'santa'),
(-0.05441219270766459, 'another'),
(-0.05355683986718789, 'cigar'),
(-0.052625991688476856, 'box'),
(-0.05247597993218918, 'harmonious'),
(-0.0521516194029579, 'aged'),
(-0.052094946152625915, 'months'),
(-0.05199773759926687, 'american'),
(-0.05193803238622454, 'sauce'),
(-0.051873162588974495, 'mushrooms'),
(-0.05178291086309458, 'try'),
(-0.05100711860414376, 'approachable'),
(-0.05075646739816733, 'bold'),
(-0.04950577467480381, 'wonderfully'),
(-0.04921140167306495, 'lamb'),
(-0.048951469379990564, 'toasted'),
(-0.04825271644340732, 'such'),
(-0.048232428672725945, 'elements'),
(-0.048080205801724445, 'cellar'),
(-0.047975591871509654, 'taste'),
(-0.047653614826910544, 'generous'),
(-0.047525288777149496, 'spiced'),
(-0.046925351601044996, 'satisfying'),
(-0.0468437789310385, 'graphite'),
(-0.04652149475066161, 'sonoma'),
(-0.046358477530556705, 'county'),
(-0.04623838878523717, 'delivers'),
(-0.04601287495631383, 'leading'),
(-0.04585792420875427, 'characteristics'),
(-0.045326918363981186, 'purity'),
(-0.045326265793954594, 'pure'),
(-0.045129418391315186, '1'),
(-0.045079010032576174, 'mild'),
(-0.044822847671085665, 'inviting'),
(-0.04434811452469254, 'spectator'),
(-0.04434811452469254, 'top'),
```

```
(-0.04430885908035076, 'lingering'),
(-0.04421359002691258, 'enjoyed'),
(-0.04409234267181774, 'powder'),
(-0.04365089927584173, 'flowers'),
(-0.04265512367319019, 'summer'),
(-0.04262018294181623, 'always'),
(-0.04246031846474393 5, 'fruit'),
(-0.04206212995224773, 'within'),
(-0.04182462915568238, 'currants'),
(-0.04172040895777861, 'milk'),
(-0.04113313533801532, 'cheese'),
(-0.041003080609227784, 'attack'),
(-0.040413454115247265, 'some'),
(-0.040294518730636095, 'between'),
(-0.04021662055388567, 'forest'),
(-0.04015587159697802, 'pleasing'),
(-0.04005913556469819, 'toasty'),
(-0.04002421959418155, 'vineyard'),
(-0.03999426638313689, 'raspberries'),
(-0.039875708961188645, 'addition'),
(-0.03891458446743527, 'blended'),
(-0.03889586037532892, 'core'),
(-0.038798993922257545, 'age'),
(-0.038753943155507675, 'pomegranate'),
(-0.03808708042792401, 'many'),
(-0.037382649010518756, 'any'),
(-0.03688787232669303, 'layers'),
(-0.036405085576181606, 'signature'),
(-0.035100704013808846, 'succulent'),
(-0.035057662066258634, 'example'),
(-0.03493624709724325, 'perfumed'),
(-0.03489546074110786, 'balance'),
(-0.03458259932184804, 'profile'),
(-0.034448426217757185, 'been'),
(-0.0343765338130128, 'blackcurrant'),
(-0.034210149654785506, 'your'),
(-0.03409084066641976, 'potential'),
(-0.03408717716190672, 'toffee'),
(-0.0340526913684669, 'characteristic'),
(-0.03380302791075587, 'best'),
(-0.033670766842117096, 'finishes'),
(-0.03347111320157789, 'even'),
(-0.03339404454973642, 'excellent'),
(-0.03325566863394517, 'chewy'),
(-0.03266543854222803, 'polished'),
(-0.032587412857111586, 'overtones'),
(-0.032585469788512664, 'note'),
(-0.03241100790696488, 'great'),
(-0.03220085022470866, 'old'),
(-0.032153660279509236, 'brown'),
(-0.0317655956162875, 'ripe'),
(-0.03154006662171574, 'classic'),
(-0.030860944834470428, 'savory'),
(-0.03065561758602314, 'baking'),
(-0.030447988516512703, '6'),
(-0.030071756796489553, 'winery'),
(-0.02998651777629209, '!'),
```

```
(-0.02997426798199811, 'bouquet'),
(-0.029748948727348873, 'toast'),
(-0.02972228990257716, 'focused'),
(-0.029269378993598952, 'boysenberry'),
(-0.028904406371283626, 'across'),
(-0.028759301877392198, 'hills'),
(-0.028730435747923572, 'so'),
(-0.02840080507186584, '%'),
(-0.028199200184550452, '2006'),
(-0.027726966162814237, 'petal'),
(-0.02722274370875296, 'perfectly'),
(-0.0268891770686951, 'floral'),
(-0.026778655166066014, 'features'),
(-0.025874652336901074, 'winemaking'),
(-0.025730657491490967, 'most'),
(-0.025515255610226512, 'bottle'),
(-0.02532408687661461, 'crushed'),
(-0.025273126312462445, 'next'),
(-0.025153360454656517, 'finely'),
(-0.024934107049472933, 'pastas'),
(-0.024883972567859243, 'clove'),
(-0.024875338162207463, 'exceptional'),
(-0.024859631330243097, 'flavorful'),
(-0.02475660668818926, 'same'),
(-0.02472993290603141, 'almonds'),
(-0.02426235021189905, 'veal'),
(-0.024232522481733285, '4'),
(-0.024187256523636944, 'sauces'),
(-0.02409657387131372, 'continue'),
(-0.024039340325218192, 'here'),
(-0.024039340325218192, 'three'),
(-0.023993488602426494, 'back'),
(-0.02397458964919466, 'enough'),
(-0.023843056941241628, 'come'),
(-0.023690718194130198, 'floor'),
(-0.02364607368243671, 'mouth-filling'),
(-0.02330105793507703, 'touch'),
(-0.02295663290883217, 'would'),
(-0.022869242529563096, 'wood'),
(-0.02284956471657004, 'mingle'),
(-0.022813359454970497, 'those'),
(-0.022807065047459708, 'reflects'),
(-0.022807065047459708, 'select'),
(-0.022795355871118105, 'range'),
(-0.022795355871118105, 'strawberries'),
(-0.022795355871118105, 'wide'),
(-0.02190573522673844, 'cinnamon'),
(-0.021681020204528313, 'roast'),
(-0.021679291216385184, 'quality'),
(-0.02105266807031885, 'style'),
(-0.020812699722644504, 'additional'),
(-0.02073043222035046, 'fine-grained'),
(-0.020207266242306256, 'our'),
(-0.019894203063000805, 'personality'),
(-0.019833027830779098, 'nicely'),
(-0.019775741301015483, 'intense'),
(-0.019676125390211897, 'conditions'),
```

```
(-0.01960703525053257, 'lavender'),
(-0.01946829428039389, 'quite'),
(-0.019422434708960903, 'provide'),
(-0.019280024632237762, 'lingers'),
(-0.01918710605555198, 'lovely'),
(-0.01909237920129735, 'blending'),
(-0.018136391455965736, 'get'),
(-0.017506200162369458, 'complex'),
(-0.016967797931704928, 'fruit-forward'),
(-0.01669663014372772, 'offering'),
(-0.016624801068122242, 'alexander'),
(-0.01652590235975547, 'during'),
(-0.016422411161821117, 'elegantly'),
(-0.016356557267851953, 'extremely'),
(-0.016356557267851953, 'off'),
(-0.01616693248235538, 'versatile'),
(-0.015727630796624575, 'unique'),
(-0.01570766721505521, 'well-structured'),
(-0.015600112855018925, 'terroir'),
(-0.01555056801565185, 'framed'),
(-0.015472264092711251, '8'),
(-0.01538647640189167, '('),
(-0.01538647640189167, ')'),
(-0.015370117973011991, 'including'),
(-0.015347157889313267, 'match'),
(-0.015214953095794997, 'almond'),
(-0.01501344339815571, 'lift'),
(-0.014743447259083909, 'from'),
(-0.01472603469570416, '...'),
(-0.014721313928228574, 'about'),
(-0.014435559348923105, 'superb'),
(-0.01430726537826319, 'cranberry'),
(-0.01385973984040681, 'mouthfeel'),
(-0.013659116636138863, 'four'),
(-0.013349467281535778, 'density'),
(-0.01326701033636246, 'pie'),
(-0.013265999747143275, 'known'),
(-0.013035890847022596, 'give'),
(-0.012908351270554268, 'i'),
(-0.012853310627955958, '40'),
(-0.012803731700341148, 'cellaring'),
(-0.012279193995450124, 'fruity'),
(-0.01224481994453356, 'array'),
(-0.011945320420517927, 'accented'),
(-0.011387983509433352, 'wild'),
(-0.01100242725797429, 'fully'),
(-0.01092827199837742, 'butter'),
(-0.010790519688999538, 'resulting'),
(-0.010726182920309272, 'up'),
(-0.01058815639111785, 'wonderful'),
(-0.010460822206829206, 'abundant'),
(-0.010285586540507756, 'contributes'),
(-0.010260304145631878, 'richly'),
(-0.010029813594453171, 'offers'),
(-0.010028112409880927, 'seamless'),
(-0.009960449334237202, 'blueberries'),
(-0.009616273510639364, 'subtle'),
```

```
(-0.00954156930027995, 'well-integrated'),
(-0.009536371345858571, 'braised'),
(-0.009536371345858571, 'tuna'),
(-0.009356668844100503, 'vintages'),
(-0.009204755131735496, 'selected'),
(-0.009203512358419576, 'grown'),
(-0.00910289790199605, 'alongside'),
(-0.009029031073230394, 'overall'),
(-0.008995031164540899, 'creating'),
(-0.008949050943978104, 'selection'),
(-0.008938527143051659, 'enjoy'),
(-0.008850611573678389, 'open'),
(-0.008487833587946966, 'typical'),
(-0.008140742375533612, 'may'),
(-0.00802080385612684, 'coast'),
(-0.007960047665933955, 'create'),
(-0.00790426059960318, 'brooding'),
(-0.007394201377343626, 'sweetness'),
(-0.007326267188672729, 'fragrant'),
(-0.00721936929629651, 'game'),
(-0.00692123621636167, 'two'),
(-0.006701944738505513, 'little'),
(-0.006610656885339241, 'immediately'),
(-0.006582647774880557, 'favorite'),
(-0.006417891105621671, 'appellation'),
(-0.006417327959004256, 'full'),
(-0.006379534055867331, 'yields'),
(-0.005919133986675381, 'natural'),
(-0.005891440131676478, 'aromatics'),
(-0.0058371996074084, 'into'),
(-0.005597270145462712, 'follow'),
(-0.0053582144342850845, 'allspice'),
(-0.005330402161842966, 'after'),
(-0.005330402161842966, 'combined'),
(-0.005330402161842966, 'displaying'),
(-0.0052938159524082064, 'acid'),
(-0.005276999965067406, 'hazelnut'),
(-0.0052490772849422005, 'mixed'),
(-0.004985496388268451, 'adds'),
(-0.004771931842155386, 'carry'),
(-0.0047447079316699226, 'lifted'),
(-0.004612317633887332, 'friendly'),
(-0.004548824567402557, 'sirah'),
(-0.004508221030662316, 'bread'),
(-0.004508221030662316, 'wet'),
(-0.004461318678245742, 'end'),
(-0.0044463005354140935, 'strong'),
(-0.004393158807715446, 'perfume'),
(-0.0043175360869620195, 'duck'),
(-0.004263689793659112, 'traditional'),
(-0.003924009268011611, 'exudes'),
(-0.003919045188010728, '15'),
(-0.003721667483051531, 'lightly'),
(-0.0036124855078937088, 'complimented'),
(-0.003264666467019741, 'bing'),
(-0.0030054098073379787, 'northern'),
(-0.0030054098073379787, 'tannic'),
```

```
(-0.002985216020023743, 'wrapped'),
(-0.002931477587003009, 'spring'),
(-0.002712332585000495, 'sandalwood'),
(-0.0021238518102777463, 'baked'),
(-0.0021238518102777463, 'creme'),
(-0.0021238518102777463, 'place'),
(-0.001985126612927801, 'include'),
(-0.0019817642513876997, 'chicken'),
(-0.001845759483543884, 'lots'),
(-0.001590528623926307, 'hard'),
(-0.001522319314498295, 'd'),
(-0.001460034948560884, 'every'),
(-0.0014321122684356792, 'attractive'),
(-0.0012379678077982504, 'well-balanced'),
(-0.0006813746009398917, 'luscious'),
(-0.0005274052564772116, 'family'),
(-0.0004656374637965058, 'complemented'),
(0.0, '30'),
(0.0, '50'),
(0.0, 'abundance'),
(0.0, 'achieve'),
(0.0, 'almost'),
(0.0, 'barossa'),
(0.0, 'bay'),
(0.0, 'bring'),
(0.0, 'brioche'),
(0.0, 'bubbles'),
(0.0, 'carneros'),
(0.0, 'champagne'),
(0.0, 'clones'),
(0.0, 'finesse'),
(0.0, 'friends'),
(0.0, 'gently'),
(0.0, 'ginger'),
(0.0, 'goes'),
(0.0, 'grenache'),
(0.0, 'impression'),
(0.0, 'mourvedre'),
(0.0, 'mousse'),
(0.0, 'mushroom'),
(0.0, 'paired'),
(0.0, 'persistence'),
(0.0, 'pizza'),
(0.0, 'produce'),
(0.0, 'sangiovese'),
(0.0, 'smoked'),
(0.0, 'sparkling'),
(0.0, 'strawberry'),
(0.0, 'tart'),
(0.0, 'thai'),
(0.0, 'what'),
(6.815092076737871e-05, 'both'),
(0.00015359507406814205, 'minerals'),
(0.0005260613317533032, 'sourced'),
(0.001062026395610098, '9'),
(0.0032920398918419367, 'mouthwatering'),
(0.0035316980449502386, 'ribs'),
```

```
(0.0037667656754334328, 'an'),
(0.0038250243765400255, 'fleshy'),
(0.0047603315746479885, 'bottling'),
(0.005783458471492601, 'candied'),
(0.005783458471492601, 'crab'),
(0.006057158691699302, 'barrels'),
(0.006196655130568278, 'perfect'),
(0.006218697362174963, 'particularly'),
(0.006537752275114267, 'feel'),
(0.006548343487952563, 'juicy'),
(0.006704499353963358, 'add'),
(0.006748692518236935, 'lighter'),
(0.007712358596566881, 'pairing'),
(0.009157280083810343, 'dominated'),
(0.009205286586966796, 'were'),
(0.00920907047453397, 'provides'),
(0.009396582631483857, 'producing'),
(0.009997489116471353, '12'),
(0.010103059274376406, 'ideal'),
(0.010164269102222252, 'aging'),
(0.010690455423614144, 'right'),
(0.010710445460833475, 'stone'),
(0.010933968120504784, 'grape'),
(0.011011825050261268, 'due'),
(0.011266959056400964, 'entry'),
(0.01178083193032755, 'quince'),
(0.012358128462236242, 'harvest'),
(0.013436710762108936, 'making'),
(0.013550146489739817, 'scents'),
(0.015101964957214246, 'finest'),
(0.01521807531352132, 'malolactic'),
(0.015445599076138634, 'clear'),
(0.015445599076138634, 'salad'),
(0.015445599076138634, 'salmon'),
(0.015738784160949225, 'flavor'),
(0.016016540505010803, 'distinct'),
(0.017564290401820143, 'honey'),
(0.0176312618374207, 'cream'),
(0.017643171091933838, 'without'),
(0.017766782928880703, 'layered'),
(0.017947400620027917, 'partner'),
(0.0183066705256871, 'mid'),
(0.019448113874987526, 'pleasant'),
(0.019521419873576692, 'notes'),
(0.019829772403973347, 'especially'),
(0.020168261776284654, 'several'),
(0.02043179112409015, 'mid-palate'),
(0.020853450364752446, 'when'),
(0.02140479610785825, 'fruits'),
(0.022032367798951 29, 'delightful'),
(0.022107237226826754, 'creek'),
(0.022343075724892567, 'food'),
(0.022989476271705778, 'enhanced'),
(0.022989476271705778, 'russian'),
(0.02365456907281494, 'petite'),
(0.023891270162542288, 'delicious'),
(0.024082355842857034, 'flavored'),
```

```
(0.024629075400698402, 'foods'),
(0.025097675954200867, 'showing'),
(0.025421139921855233, 'aftertaste'),
(0.02553897748150865, 'accompaniment'),
(0.025608763707235933, 'intensely'),
(0.026213434042521407, '100'),
(0.028606765833250194, 'no'),
(0.028749988660768125, 'own'),
(0.028884030821991712, 'golden'),
(0.028884030821991712, 'pink'),
(0.029799228970101066, 'creamy'),
(0.02991242013950006, 'orange'),
(0.02991242013950006, 'peel'),
(0.030081589382208916, 'brilliant'),
(0.03048892510460423, 'along'),
(0.03095428474139889, 'nose'),
(0.031623066395208406, 'high'),
(0.03188636144516659, 'pairs'),
(0.032499307558750745, 'peaches'),
(0.03313830196805763, 'central'),
(0.03370167869300179, 'underlying'),
(0.033881999125718404, 'barrel'),
(0.03404528206360848, 'exotic'),
(0.03407134772303762, 'edge'),
(0.03714345748765705, 'middle'),
(0.03728699672585595, 'appealing'),
(0.03741143129606554, 'dried'),
(0.037448490276430116, 'delicate'),
(0.03855109169961687, 'then'),
(0.03856461972835974, '2010'),
(0.03865495640667692, 'fish'),
(0.03908866404384217, 'jasmine'),
(0.03928668985675452, 'combine'),
(0.03947073648005803, 'out'),
(0.0418586322913377, 'season'),
(0.0420773649316998, 'accompanied'),
(0.0430584104895629, 'elegance'),
(0.043236838986748224, 'aromatic'),
(0.04328821498476485, 'where'),
(0.045190853015372234, 'persistent'),
(0.045212306042342346, '-'),
(0.04534356486265953, 'true'),
(0.04583323682083263, 'grapes'),
(0.04660403218295014, 'that'),
(0.04712526678562763, 'distinctive'),
(0.047784991897353545, 'leads'),
(0.04863386544139489, 'asian'),
(0.04863386544139489, 'served'),
(0.04863386544139489, 'slightly'),
(0.048898414781682574, 'as'),
(0.050914355442321944, 'la'),
(0.05160455974693019, 'rock'),
(0.051721498706325565, 'linger'),
(0.053557906519173554, 'combination'),
(0.05380782594828921, 'was'),
(0.05516544344046728, 'enjoyable'),
(0.056063677151666574, 'brings'),
```

```
(0.05820453779773626, 'showcases'),
(0.05866033744272786, 'apple'),
(0.05915088877181146, 'truly'),
(0.06113601538473927, 'love'),
(0.0613781357154096, 'fig'),
(0.06143024089429863, 'their'),
(0.062089615300345236, 'poultry'),
(0.06231847293897558, 'freshness'),
(0.0627177228024165, 'also'),
(0.06277721774972746, 'have'),
(0.0631486113894493, 'salads'),
(0.06429955116976223, ':'),
(0.06460191572139116, 'qualities'),
(0.06485157165605086, 'made'),
(0.06515572008879993, 'supported'),
(0.06533517798591938, 'undertones'),
(0.06735267589989864, 'makes'),
(0.06763003710157266, 'there'),
(0.06838915735802126, 'cuisine'),
(0.06844836322586298, 'leaf'),
(0.0687309816213181, 'comes'),
(0.06987346288164666, 'adding'),
(0.07098129713200528, 'touches'),
(0.07173972428144006, 'us'),
(0.07189817640780038, 'driven'),
(0.07351963400642268, 'combines'),
(0.07351963400642268, 'lees'),
(0.07511232833316685, 'honeydew'),
(0.07519642670078015, 'tones'),
(0.07819336482774539, 'small'),
(0.07836849247147148, 'honeysuckle'),
(0.07877914714547002, 'result'),
(0.07893933389836019, 'low'),
(0.0804465435800802, 'south'),
(0.0812154516650806, 'mature'),
(0.08137230635179636, 'climate'),
(0.08342913365115401, 'length'),
(0.08434579498673664, 'freshly'),
(0.08520115904947774, 'hues'),
(0.08647569846495498, 'exhibits'),
(0.08815229999955704, 'are'),
(0.08906997623818076, 'ample'),
(0.08924877415259312, 'weight'),
(0.09023777411615568, 'butterscotch'),
(0.09071578357277055, 's'),
(0.09080086625165211, 'apples'),
(0.09150858356216217, 'flavours'),
(0.09151678628372849, 'by'),
(0.09187167478314745, 'pears'),
(0.09235606373703138, 'solid'),
(0.0929100797905534, 'shellfish'),
(0.09324654536623356, 'glass'),
(0.0939708595189714, 'complement'),
(0.09400365459980907, 'blossom'),
(0.09561697199160168, 'displays'),
(0.09569809138293717, 'together'),
(0.09738991467177886, 'alcohol'),
```

```
(0.0994140990760474, 'richness'),
(0.10066148884665636, 'cooler'),
(0.10066148884665636, 'produced'),
(0.10138135195602284, 'way'),
(0.10173247767124029, 'complexity'),
(0.10178388209529264, 'gold'),
(0.10215350323690378, 'highlights'),
(0.10253085274113911, 'color'),
(0.10336527246023634, 'refined'),
(0.10368890334342509, 'complements'),
(0.10427050318096358, 'gives'),
(0.10528364595341891, 'giving'),
(0.10541561973562216, 'backbone'),
(0.1063826227035687, 'varieties'),
(0.10862009659229802, 'passionfruit'),
(0.10936359249740983, 'reminiscent'),
(0.1107718593667633, 'through'),
(0.11223582453880325, 'they'),
(0.11588029442117083, '7'),
(0.11950983959235137, 'herbs'),
(0.12124119834673404, 'lengthy'),
(0.12277693950597346, 'aperitif'),
(0.12589233932950294, 'clean'),
(0.1259979940807042, 'nectarine'),
(0.1276893007091012, 'dishes'),
(0.1278154421419027, 'character'),
(0.12822113738636734, 'cut'),
(0.12853180591293292, 'good'),
(0.12877182183639843, 'growing'),
(0.13099224904336615, 'created'),
(0.1323208021661741, 'river'),
(0.13544227276133552, 'this'),
(0.13596266777282517, 'aromas'),
(0.14095935399786153, 'characters'),
(0.14097782262717498, 'lively'),
(0.14278089312246933, 'stainless'),
(0.14278089312246933, 'steel'),
(0.1429442043611824, 'tangerine'),
(0.14409713689465503, 'it'),
(0.14433195682433209, 'chilled'),
(0.14495128028672064, 'mix'),
(0.14618464702175568, 'only'),
(0.14776766974653338, 'seafood'),
(0.15018177368491892, 'serve'),
(0.15666674731446126, 'herbal'),
(0.1567985600250197, 'fermented'),
(0.16449555304741537, 'intensity'),
(0.16621791198811464, 'guava'),
(0.16677763637992776, 'apricot'),
(0.16799101920546766, 'added'),
(0.17037901166206684, 'white'),
(0.17806537499011357, 'lychee'),
(0.18134608640343702, 'yellow'),
(0.18936001348798487, 'not'),
(0.19367597251528618, 'bright'),
(0.19741488328407417, 'cool'),
(0.20329245526064443, 'oysters'),
```

```
(0.2069087921220055, '/'),
(0.206996340920679, 'variety'),
(0.21130818762732428, '2011'),
(0.22707170907476454, 'well'),
(0.2286583228556483, 'palate'),
(0.23295023363033568, 'itself'),
(0.23570837854761004, 'possesses'),
(0.23695982922082828, 'pineapple'),
(0.2450946058282437, 'while'),
(0.24706902548100504, 'grass'),
(0.24960398355973254, 'fermentation'),
(0.25075147061512726, 'zest'),
(0.2633273055891914, 'vibrant'),
(0.27002315419864004, 'pear'),
(0.29521312858217924, 'peach'),
(0.31627825014003513, 'zesty'),
(0.3177883882838997, 'mineral'),
(0.31996351363416037, 'meyer'),
(0.3297623686922546, 'straw'),
(0.33179437888735736, 'pale'),
(0.3541340373864262, 'light'),
(0.3645903613724852, 'passion'),
(0.4313024748721522, 'fresh'),
(0.45171278657983177, 'green'),
(0.48047640021777205, 'refreshing'),
(0.4950979194739073, 'minerality'),
(0.5068182594089797, 'lemon'),
(0.5120902639707063, 'gooseberry'),
(0.5599900369601929, 'grapefruit'),
(0.5842287065766666, 'melon'),
(0.6064232619016869, 'tropical'),
(0.6891990079029966, 'acidity'),
(0.7108905521536927, 'lime'),
(0.7701247691557761, 'crisp'),
(0.8832897531176555, 'citrus')]
```

WU5 Logistic Loss

Logistic Loss performs the best

Top 5 positive

1. citrus
2. crisp
3. lime
4. acidity
5. tropical

Top 5 negative

1. tannins
2. black
3. dark
4. cherry
5. blackberry

From the code run above, it is shown the weight of each word. With lowest weight reach -1.1 and the highest reach 0.88

In [ ]:
```
f = linear.LinearClassifier({'lossFunction': linear.HingeLoss(), 'lambda'
runClassifier.trainTestSet(f, datasets.WineDataBinary)
```

```
Training accuracy 0.753036, test accuracy 0.686347
```

In [ ]:
```

```