

# ML LAB-1

## EXPT #1: Creating attributes and test and train sets

1. 1000 random examples were chosen from TRAINlabeledBow.feats and 1000 from TESTlabeledBow.feats such that half of them are positive and half are negative.
2. For attributes, I chose 5000 most frequently used words with 2500 having polarity>1 and 2500 with polarity<-0.5.
3. For different runs, the examples in train and test sets change as they are chosen randomly and so, the accuracy also changes by a slight amount

## EXPT #2: ID3 with and without early stopping

Without early stopping:

Accuracy(avg) : 69.5%

Num of leaves: 189

5 most used attributes(for splitting)	Frequency
53	7
88	4
115	4
313	3
413	3

After early stopping:

Accuracy: 70.19%

Num of leaves: 119

5 most used attributes(for splitting)	Frequency
156	3
115	3
297	3
83	2
856	2

```
522         temp = temp.right
523     else:
524         if temp.label=="yes":
525             myans = "yes"
526         elif temp.label=="no":
527             myans = "no"
528         if myans==ans:
529             totalcorrect = totalcorrect + 1
530         break
531
532 correctPercent = totalcorrect/total
533 correctPercent = correctPercent*100
534 print("\nid3 after early stopping(stopped at num of examples in a node<30):
535 print("\num of leaves: ",root.count_leaf_node)
536 dict_attr_freq = root.count_attr_freq()
537 i = 0
538 print("\n5 most used attributes to split are(
539 for key in sorted(dict_attr_freq.items(), key=
540 i = i+1
541 print(key[0], "\t", key[1])
542 if i>4:
543     break
544 totalcorrect = 0
545 lines = [line.rstrip('\n') for line in open("train.txt")]
546 total = len(lines)
547 for line in lines:
548     l = line.split(" ")
549     rating = int(l[0])
```

```
sahil@sahil-VirtualBox:~/ML/lab1$ cd soln/
sahil@sahil-VirtualBox:~/ML/lab1/soln$ cd 2016csb1057lab1/
sahil@sahil-VirtualBox:~/ML/lab1/soln/2016csb1057lab1$ python3 id3.py test.txt 2
id3 without early stopping:
Accuracy: 69.5
num of leaves: 189
5 most used attributes to split are(attributeIndex numOfTimesUsed):
53 7
88 4
115 4
313 3
413 3
id3 after early stopping(stopped at num of examples in a node<30):
Accuracy: 70.19999999999999
num of leaves: 119
5 most used attributes to split are(attributeIndex numOfTimesUsed):
156 3
115 3
297 3
83 2
856 2
Accuracy on training set after early stopping: 92.4
sahil@sahil-VirtualBox:~/ML/lab1/soln/2016csb1057lab1$
```

So, as we can see early stopping increases the accuracy by a percent as the size of tree decreases and thus the overfitting also decreases. But the stopping criteria must not be too high, in my case it was 30 examples in a node.

Too high or too low stopping criteria decreased the accuracy.

### EXPT #3: ID3 with noisy examples:

Noise(in percent)	Accuracy
0.5%	70%
1%	69.19%
5%	68.5%
10%	65%

LINUX-UBUNTU [Running] - Oracle VM VirtualBox

```
File Machine View Input Devices Help
```

Terminal

```
570 temp = temp.right
571
572 else:
573     if temp.label=="yes":
574         myans = "yes"
575     elif temp.label=="no":
576         myans = "no"
577     if myans==ans:
578         totalcorrect = totalcorrect + 1
579         break
580 correctPercent = totalcorrect/total
581 correctPercent = correctPercent*100
582 print("Accuracy on training set after early stop")
583 elif exptno == '3':
584     import random
585     noise = 5 #noisy elements
586     lines = [line.rstrip('\n') for line in open('train.txt')]
587     d = {}
588     i = 1
589     for line in lines:
590         d[i] = line
591         i = i + 1
592     for i in range(0,noise):
593         a = random.randint(1,1000)
594         if d[a][0]=='1' and d[a][1]=='0':
595             string = "1"
596             string = string + d[a][2:]
597             d[a] = string
```

sahil@sahil-VirtualBox: ~/ML/lab1/soln/2016csb1057lab1

```
sahil@sahil-VirtualBox:~/ML/lab1/soln/2016csb1057lab1$ python3 id3
.py test.txt 3
Effect of 1.0 % noise
Accuracy: 69.19999999999999
sahil@sahil-VirtualBox:~/ML/lab1/soln/2016csb1057lab1$ python3 id3
.py test.txt 3
Effect of 0.5 % noise
Accuracy: 70.0
sahil@sahil-VirtualBox:~/ML/lab1/soln/2016csb1057lab1$
```

Line 585, Column 14

Tab Size: 4 Python

02:09 PM 28-08-2018

LINUX-UBUNTU [Running] - Oracle VM VirtualBox

```
File Machine View Input Devices Help
```

Terminal

```
580 correctPercent = totalcorrect/total
581 correctPercent = correctPercent*100
582 print("Accuracy on training set after early stopping: ", correctPercent)
583 elif exptno == '3':
584     import random
585     noise = 50 #noisy elements
586     lines = [line.rstrip('\n') for line in open('train.txt')]
587     d = {}
588     i = 1
589     for line in lines:
590         d[i] = line
591         i = i + 1
592     for i in range(0,noise):
593         a = random.randint(1,1000)
594         if d[a][0]=='1' and d[a][1]=='0':
595             string = "1"
596             string = string + d[a][2:]
597             d[a] = string
598         elif int(d[a][0])>=7:
599             string = "1"
600             string = string + d[a][1:]
601             d[a] = string
602         elif int(d[a][0])<=4:
603             string = "9"
604             string = string + d[a][1:]
605             d[a] = string
```

sahil@sahil-VirtualBox: ~/ML/lab1/soln/2016csb1057lab1

```
sahil@sahil-VirtualBox:~/ML/lab1/soln/2016csb1057lab1$ python3 id3
.py test.txt 3
Effect of 10.0 % noise
Accuracy: 66.4
sahil@sahil-VirtualBox:~/ML/lab1/soln/2016csb1057lab1$ python3 id3
.py test.txt 3
Effect of 5.0 % noise
Accuracy: 68.5
sahil@sahil-VirtualBox:~/ML/lab1/soln/2016csb1057lab1$
```

noise

4 matches

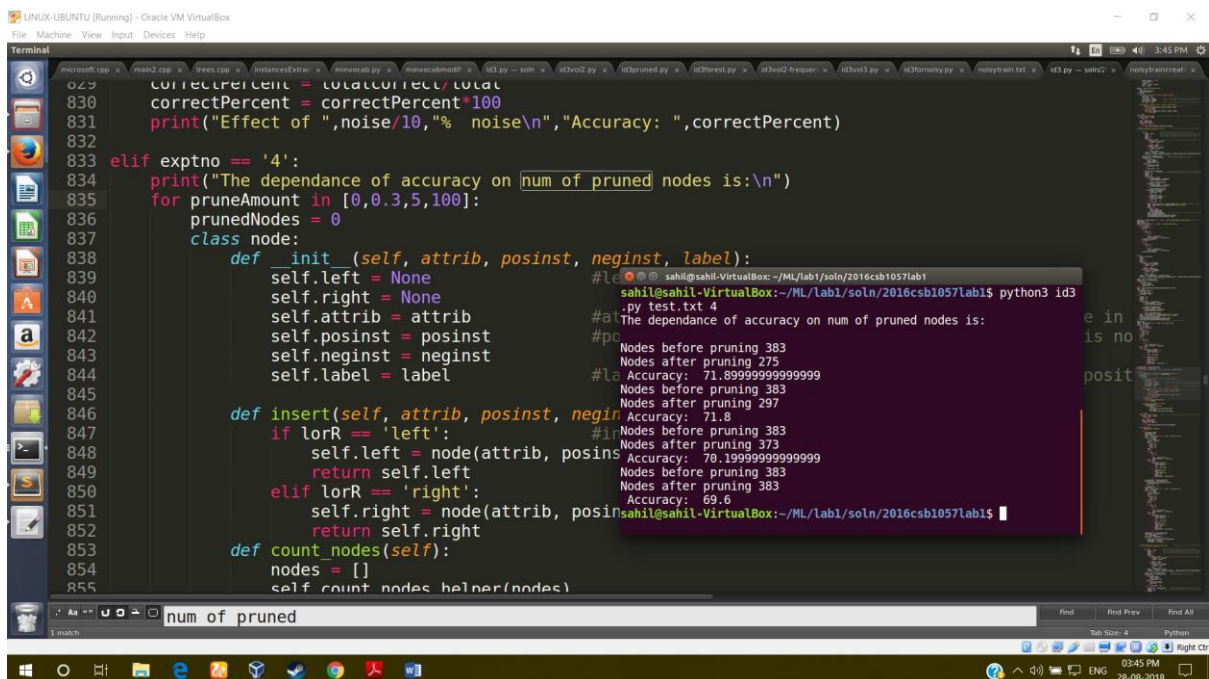
Tab Size: 4 Python

01:52 PM 28-08-2018

## EXPT #4: ID3 with post-pruning(reduced error pruning)

Nodes before pruning	Nodes after pruning	Accuracy
383	275	71.9
383	297	71.8%
383	383(no pruning)	69.6%

So, the number of nodes pruned directly affects the accuracy, if a lot of nodes are pruned then the accuracy decreases but up until a level the accuracy increases by upto 2% on pruning



```
correctPercent = totalCorrect/total
correctPercent = correctPercent*100
print("Effect of ",noise/10,"% noise\n","Accuracy: ",correctPercent)

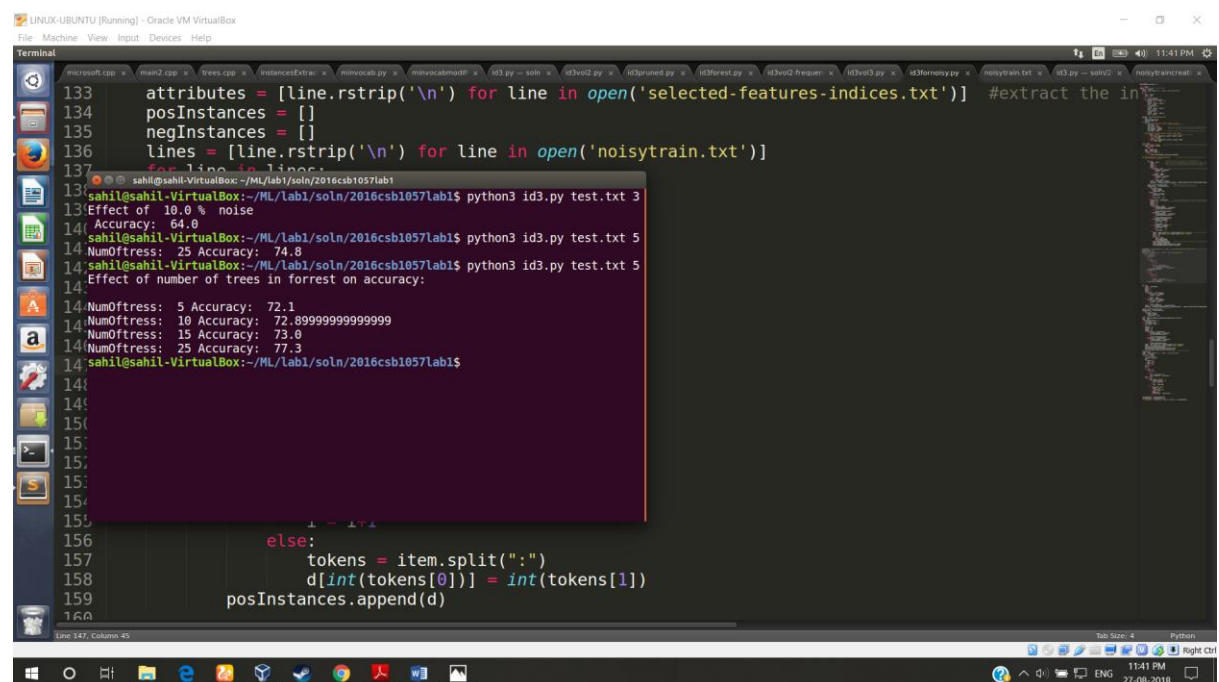
elif exptno == '4':
    print("The dependance of accuracy on num of pruned nodes is:\n")
    for pruneAmount in [0,0.3,5,100]:
        prunedNodes = 0
        class node:
            def __init__(self, attrib, posinst, neginst, label):
                self.left = None
                self.right = None
                self.attrib = attrib
                self.posinst = posinst
                self.neginst = neginst
                self.label = label
            def insert(self, attrib, posinst, neginst, label):
                if lorR == 'left':
                    self.left = node(attrib, posinst, neginst, label)
                    return self.left
                elif lorR == 'right':
                    self.right = node(attrib, posinst, neginst, label)
                    return self.right
            def count_nodes(self):
                nodes = []
                self.count_nodes_helper(nodes)

#le
sahil@sahil-VirtualBox:~/ML/lab1/soln/2016csb1057lab1
sahil@sahil-VirtualBox:~/ML/lab1/soln/2016csb1057lab1$ python3 id3
.py test.txt 4
#a
#p
The dependance of accuracy on num of pruned nodes is:
Nodes before pruning 383
Nodes after pruning 275
Accuracy: 71.89999999999999
Nodes before pruning 383
Nodes after pruning 297
Accuracy: 71.8
Nodes before pruning 383
Nodes after pruning 373
Accuracy: 70.19999999999999
Nodes before pruning 383
Nodes after pruning 383
Accuracy: 69.6
sahil@sahil-VirtualBox:~/ML/lab1/soln/2016csb1057lab1$
```

## EXPT #5: Random forests using feature bagging

Randomly 2000 attributes were chosen from the list of 5000 attributes and then a number of trees were created and the maximum output of those trees was chosen as my output.

Number of trees	Accuracy
5	72.1%
10	72.8%
15	73%
25	77.3%



```
133 attributes = [line.rstrip('\n') for line in open('selected-features-indices.txt')] #extract the in
134 posInstances = []
135 negInstances = []
136 lines = [line.rstrip('\n') for line in open('noisytrain.txt')]
137 for line in lines:
138     sahil@sahil-VirtualBox: ~/ML/lab1/soln/2016csb1057lab1
139 sahil@sahil-VirtualBox:~/ML/lab1/soln/2016csb1057lab1$ python3 id3.py test.txt 3
140 Effect of 10.0 % noise
141 Accuracy: 64.0
142 sahil@sahil-VirtualBox:~/ML/lab1/soln/2016csb1057lab1$ python3 id3.py test.txt 5
143 Effect of number of trees in forrest on accuracy:
144 NumOftrss: 5 Accuracy: 72.1
145 NumOftrss: 10 Accuracy: 72.89999999999999
146 NumOftrss: 15 Accuracy: 73.0
147 NumOftrss: 25 Accuracy: 77.3
148 sahil@sahil-VirtualBox:~/ML/lab1/soln/2016csb1057lab1$
149
150
151
152
153
154
155
156 else:
157     tokens = item.split(":")
158     d[int(tokens[0])] = int(tokens[1])
159 posInstances.append(d)
160
```