



Blog

🏠 > Machine Learning Lab > 3. AI AND MACHINE LEARNING VTU LAB | READ NOW

3. AI AND MACHINE LEARNING VTU LAB | READ NOW

👤 VTULOOP - 🕒 October 20, 2021 - 📁 Machine Learning Lab - 💬 0 Comments

MACHINE LEARNING VTU LAB

Program -3] WRITE A PROGRAM TO DEMONSTRATE THE WORKING OF THE DECISION TREE BASED ID3 ALGORITHM. USE AN APPROPRIATE DATA SET FOR BUILDING THE DECISION TREE AND APPLY THIS KNOWLEDGE TO CLASSIFY A NEW SAMPLE.

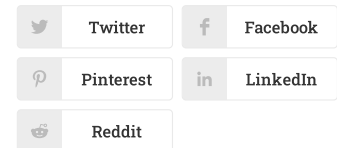
Program Code- lab3.py

```
1. import numpy as np
2. import math
3. import csv
4.
5. def read_data(filename):
6.     with open(filename, 'r') as csvfile:
7.         datareader = csv.reader(csvfile, delimiter=',')
8.         headers = next(datareader)
9.         metadata = []
10.        traindata = []
11.        for name in headers:
12.            metadata.append(name)
13.        for row in datareader:
14.            traindata.append(row)
15.
16.        return (metadata, traindata)
17.
18. class Node:
19.     def __init__(self, attribute):
20.         self.attribute = attribute
21.         self.children = []
22.         self.answer = ""
23.
```



Build Your Brand Online
Squarespace

Please Share This



Newsletter

Get all latest content delivered to your email!

Your Email

Manage Your Unstructured Data - Visualize Unstructured Data

A Smarter Data Preparation Process Begins with Dataloop End to End Platform. Start Free [dataloop.ai](#)

APRI

```

30. count = np.zeros((items.shape[0], 1), dtype=np.int32)
31.
32. for x in range(items.shape[0]):
33.     for y in range(data.shape[0]):
34.         if data[y, col] == items[x]:
35.             count[x] += 1
36.
37. for x in range(items.shape[0]):
38.     dict[items[x]] = np.empty((int(count[x]), data.shape[1]), dtype="|S32")
39.     pos = 0
40.     for y in range(data.shape[0]):
41.         if data[y, col] == items[x]:
42.             dict[items[x]][pos] = data[y]
43.             pos += 1
44.     if delete:
45.         dict[items[x]] = np.delete(dict[items[x]], col, 1)
46.
47. return items, dict
48.
49. def entropy(S):
50.     items = np.unique(S)
51.
52.     if items.size == 1:
53.         return 0
54.
55.     counts = np.zeros((items.shape[0], 1))
56.     sums = 0
57.
58.     for x in range(items.shape[0]):
59.         counts[x] = sum(S == items[x]) / (S.size * 1.0)
60.
61.     for count in counts:
62.         sums += -1 * count * math.log(count, 2)
63.     return sums
64.
65. def gain_ratio(data, col):
66.     items, dict = subtables(data, col, delete=False)
67.
68.     total_size = data.shape[0]
69.     entropies = np.zeros((items.shape[0], 1))
70.     intrinsic = np.zeros((items.shape[0], 1))
71.
72.     for x in range(items.shape[0]):
73.         ratio = dict[items[x]].shape[0]/(total_size * 1.0)
74.         entropies[x] = ratio * entropy(dict[items[x]][:, -1])
75.         intrinsic[x] = ratio * math.log(ratio, 2)
76.
77.     total_entropy = entropy(data[:, -1])
78.     iv = -1 * sum(intrinsic)
79.
80.     for x in range(entropies.shape[0]):
81.         total_entropy -= entropies[x]
82.
83.     return total_entropy / iv
84.
85. def create_node(data, metadata):
86.     if (np.unique(data[:, -1])).shape[0] == 1:
87.         node = Node("")
88.         node.answer = np.unique(data[:, -1])[0]
89.         return node
90.
91.     gains = np.zeros((data.shape[1] - 1, 1))
92.
93.     for col in range(data.shape[1] - 1):
94.         gains[col] = gain_ratio(data, col)
95.
96.     split = np.argmax(gains)
97.
98.     node = Node(metadata[split])
99.     metadata = np.delete(metadata, split, 0)
100.
101.     items, dict = subtables(data, split, delete=True)
102.
103.     for x in range(items.shape[0]):
104.         child = create_node(dict[items[x]], metadata)
105.         node.children.append((items[x], child))
106.
107.     return node
108.
109. def empty(size):
110.     s = ""
111.     for x in range(size):
112.         s += " "
113.     return s

```

Recent Posts



Zoho Computer Programming
Questions and Answers

MARCH 14, 2022 / 0 COMMENTS



Apriori Algorithm | Read Now

FEBRUARY 23, 2022 / 0 COMMENTS



Association Rule Learning | Read Now

FEBRUARY 23, 2022 / 0 COMMENTS



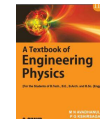
Hierarchical Clustering | Read Now

FEBRUARY 21, 2022 / 0 COMMENTS

Product-Minded Transformation
w/ Agile Software Development Services

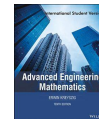
Nexient
an NIT DATA Company

[Learn More](#)



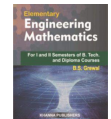
Engineering Physics by M N Avadhanulu and P G Kshirsagar

Buy Product



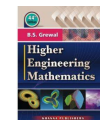
Advanced Engineering Mathematics 10ed by Erwin Krewzsig

Buy Product



Elementary Engineering Mathematics (For I & II Semesters of B. Tech. & Diploma Courses)

Buy Product



Higher Engineering Mathematics Paperback 1 by B.S.Grewel

Buy Product

How to Sell Online

SQUARESPACE

HOW TO SET UP SHOP WITH SQUARESPACE

Open [Learn More](#)

```

120.     for value, n in node.children:
121.         print(empty(level + 1), value)
122.         print_tree(n, level + 2)
123.
124. metadata, traindata = read_data("tennisdata.csv")
125. data = np.array(traindata)
126. node = create_node(data, metadata)
127. print_tree(node, 0)

```

MACHINE LEARNING Program Execution – lab3.ipynb

Jupyter Notebook program execution.

```

1. import numpy as np
2. import math
3. import csv

```

```

1. def read_data(filename):
2.     with open(filename, 'r') as csvfile:
3.         datareader = csv.reader(csvfile, delimiter=',')
4.         headers = next(datareader)
5.         metadata = []
6.         traindata = []
7.         for name in headers:
8.             metadata.append(name)
9.         for row in datareader:
10.            traindata.append(row)
11.
12.     return (metadata, traindata)

```

```

1. class Node:
2.     def __init__(self, attribute):
3.         self.attribute = attribute
4.         self.children = []
5.         self.answer = ""
6.
7.     def __str__(self):
8.         return self.attribute

```

```

1. def subtables(data, col, delete):
2.     dict = {}
3.     items = np.unique(data[:, col])
4.     count = np.zeros((items.shape[0], 1), dtype=np.int32)
5.
6.     for x in range(items.shape[0]):
7.         for y in range(data.shape[0]):
8.             if data[y, col] == items[x]:
9.                 count[x] += 1
10.
11.     for x in range(items.shape[0]):
12.         dict[items[x]] = np.empty((int(count[x]), data.shape[1]), dtype="|S32")
13.         pos = 0
14.         for y in range(data.shape[0]):
15.             if data[y, col] == items[x]:
16.                 dict[items[x]][pos] = data[y]
17.                 pos += 1
18.         if delete:
19.             dict[items[x]] = np.delete(dict[items[x]], col, 1)
20.
21.     return items, dict

```

```

1. def entropy(S):
2.     items = np.unique(S)
3.
4.     if items.size == 1:
5.         return 0
6.
7.     counts = np.zeros((items.shape[0], 1))
8.     sums = 0
9.
10.    for x in range(items.shape[0]):
11.        counts[x] = sum(S == items[x]) / (S.size * 1.0)
12.
13.    for count in counts:
14.        sums += -1 * count * math.log(count, 2)
15.    return sums

```

```

1. def gain_ratio(data, col):
2.     items, dict = subtables(data, col, delete=False)
3.
4.     total_size = data.shape[0]
5.     entropies = np.zeros((items.shape[0], 1))
6.     intrinsic = np.zeros((items.shape[0], 1))

```

```

13. total_entropy = entropy(data[:, -1])
14. iv = -1 * sum(intrinsic)
15.
16. for x in range(entropies.shape[0]):
17.     total_entropy -= entropies[x]
18.
19. return total_entropy / iv

```

```

1. def create_node(data, metadata):
2.     if (np.unique(data[:, -1])).shape[0] == 1:
3.         node = Node("")
4.         node.answer = np.unique(data[:, -1])[0]
5.         return node
6.
7.     gains = np.zeros((data.shape[1] - 1, 1))
8.
9.     for col in range(data.shape[1] - 1):
10.        gains[col] = gain_ratio(data, col)
11.
12.    split = np.argmax(gains)
13.
14.    node = Node(metadata[split])
15.    metadata = np.delete(metadata, split, 0)
16.
17.    items, dict = subtables(data, split, delete=True)
18.
19.    for x in range(items.shape[0]):
20.        child = create_node(dict[items[x]], metadata)
21.        node.children.append((items[x], child))
22.
23.    return node

```

```

1. def empty(size):
2.     s = ""
3.     for x in range(size):
4.         s += "  "
5.     return s
6.
7. def print_tree(node, level):
8.     if node.answer != "":
9.         print(empty(level), node.answer)
10.        return
11.        print(empty(level), node.attribute)
12.        for value, n in node.children:
13.            print(empty(level + 1), value)
14.            print_tree(n, level + 2)

```

```

1. metadata, traindata = read_data("tennisdata.csv")
2. data = np.array(traindata)
3. node = create_node(data, metadata)
4. print_tree(node, 0)

```

Outlook

Overcast

b'Yes'

Rainy

Windy

b'False'

b'Yes'

b'True'

b'No'

Sunny

Humidity

b'High'

b'No'

b'Normal'

b'Yes'

Alternative – LAB 3 Alt.ipynb

```

1. # Import necessary libraries
2. import pandas as pd

```

```
2. data = pd.read_csv('tennisdata.csv')
3. print("The first 5 values of data is \n",data.head())
```

The first 5 values of data is

Outlook Temperature Humidity Windy PlayTennis

0 Sunny Hot High False No

1 Sunny Hot High True No

2 Overcast Hot High False Yes

3 Rainy Mild High False Yes

4 Rainy Cool Normal False Yes

```
1. # Obtain Train data and Train output
2. X = data.iloc[:, :-1]
3. print("\nThe first 5 values of Train data is \n",X.head())
```

The first 5 values of Train data is

Outlook Temperature Humidity Windy

0 Sunny Hot High False

1 Sunny Hot High True

2 Overcast Hot High False

3 Rainy Mild High False

4 Rainy Cool Normal False

```
1. y = data.iloc[:, -1]
2. print("\nThe first 5 values of Train output is \n",y.head())
```

The first 5 values of Train output is

0 No

1 No

2 Yes

3 Yes

4 Yes

Name: PlayTennis, dtype: object

```
1. # Convert them in numbers
2. le_outlook = LabelEncoder()
3. X.Outlook = le_outlook.fit_transform(X.Outlook)
4.
5. le_Temperature = LabelEncoder()
6. X.Temperature = le_Temperature.fit_transform(X.Temperature)
7.
8. le_Humidity = LabelEncoder()
9. X.Humidity = le_Humidity.fit_transform(X.Humidity)
10.
11. le_Windy = LabelEncoder()
12. X.Windy = le_Windy.fit_transform(X.Windy)
13. print("\nNow the Train data is",X.head())
```

Now the Train data is Outlook Temperature Humidity Windy

0 2 1 0 0

1 2 1 0 1

2 0 1 0 0

3 1 2 0 0

4 1 0 1 0

```
1. le_PlayTennis = LabelEncoder()
2. y = le_PlayTennis.fit_transform(y)
3. print("\nNow the Train data is\n",y)
```

Now the Train data is

[0 0 1 1 1 0 1 0 1 1 1 1 1 0]

```

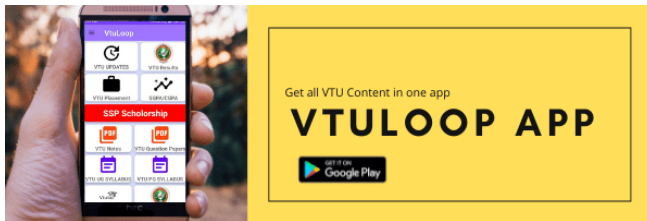
6.  ## Function to encode input
7.  def labelEncoderForInput(list1):
8.      list1[0] = le_outlook.transform([list1[0]])[0]
9.      list1[1] = le_Temperature.transform([list1[1]])[0]
10.     list1[2] = le_Humidity.transform([list1[2]])[0]
11.     list1[3] = le_Windy.transform([list1[3]])[0]
12.     return [list1]
13.
14.  ## predict for an input
15.  inp = ["Rainy","Mild","High","False"]
16.  inp1=["Rainy","Cool","High","False"]
17.  pred1 = labelEncoderForInput(inp1)
18.  y_pred = classifier.predict(pred1)
19.  y_pred
20.
21.  print("\nfor input {0}, we obtain {1}".format(inp1, le_PlayTennis.inverse_transform(y

```

for input [1, 0, 0, 0], we obtain Yes

Download the dataset

Dataset



TAGS: ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING, VTU CSE LAB



← Previous Post

2. AI AND MACHINE LEARNING VTU LAB | READ NOW

Next Post →

4. AI AND MACHINE LEARNING VTU LAB | READ NOW

➤ YOU MIGHT ALSO LIKE

Machine Learning AO* Algorithm | Read Now

2. AI AND MACHINE LEARNING VTU LAB | READ

Leave a Reply

Your comment here...

Name (required)

Email (required)

Website

☐ Save my name, email, and website in this browser for the next time I comment.

POST COMMENT

Contact Info

VTULOOP shares the free digital notes, Text books, Tutorials, Tools, etc for engineering Students.



Address:
Bangalore India



Phone:
not available



Email:
support@vtuloop.com



Website:
VTULOOP.COM

Recent Posts



**Zoho Computer
Programming
Questions and
Answers**

MARCH 14, 2022 / 0 COMMENTS



**Apriori Algorithm |
Read Now**

FEBRUARY 23, 2022 / 0 COMMENTS



**Association Rule
Learning | Read
Now**

FEBRUARY 23, 2022 / 0 COMMENTS

Store

- ☐ Women
- ☐ Men
- ☐ Brands A - Z
- ☐ Latest fashion arrivals

Useful Links

- ☐ About Us
- ☐ Contact Us
- ☐ Privacy Policy
- ☐ Disclaimer

Tags

1ST AND 2ND SEM TEXTBOOK

ENGINEERING TEXTBOOKS

MATHEMATICS TEXTBOOK

PHYSICS TEXTBOOK

Follow Us

