MACHINE, DATA AND LEARNING

Anishka Sachdeva
2018101112

Assignment 4 (Decision Tree)

Problem Statement (Question 3)

After watching a movie, your TA is highly interested in determining if the sniper shot is going to make a kill or not. He is bad at physics but good at ML. Hence to confirm his hypothesis, he asks you to come up with a decision tree which fits the below data very well.

Original Dataset

Horizontal Angle(degree)	**Distance(m)**	**Wind Speed(mph)**	**Kill**
1.5	450	220	N
4.5	520	-120	Y
3	490	120	Y
5.5	530	117	N
3.2	470	-170	N
5.2	505	-90	Y
1.85	465	120	Y
4.8	517	147	Y
1.7	430	-100	Y

Flipped indices : 2 and 4

New Dataset

Horizontal Angle(degree)	Distance(m)	Wind Speed(mph)	Kill
1.5	450	220	N
4.5	520	-120	**N**
3	490	120	Y
5.5	530	117	**Y**
3.2	470	-170	N
5.2	505	-90	Y
1.85	465	120	Y
4.8	517	147	Y
1.7	430	-100	Y

FORMULAE USED FOR THE ALGORITHM FOLLOWED

Formulas -

Entropy(E) = $- P(Y)*\log_2(P(Y)) - P(N)*\log_2(P(N))$

where P(x) is the probability of x

Remainder = $n_1/n * E_1 + n_2/n * E_2 + n_3/n * E_3 \ldots$

where E_i is the entropy of the i^{th} child of the node we want to calculate Remainder for,n_i is the entropy of the i^{th} child and n is the total entries in parent.

Information Gain = E(parent) - Remainder

APPROACH TO MAKE THE DECISION TREE

In the dataset for my question, I have all the attributes as continuous. Thus we cannot have split points which divide them into discrete sets. For our case, we can have any number of split points (<=unique values of a particular attribute) depending on the approach we use. But we had the constraint of having a split point which led to two intervals(2 children for each node) i.e. if our split point is x, then two branches forming the two intervals will be <x and >=x or <=x and >x.

To find the appropriate split point,I used the **greedy approach**. I considered each data value in each attribute to be a valid split point and calculated the respective remainders and gain values. The attribute which gave the maximum gain value was chosen to be a split point for that particular node.

Since i had to check for 18 gain values for each attribute and hence 54 gain values for all the three attributes (for the root node case), i coded up the algorithm to generate the entropy and gain values.

Shown below is the code for the same :

```python
import numpy as np
import math
num_of_features = 3
f=[
            [1.5,450,220,'N'],
            [4.5,520,-120,'N'],
            [3,490,120,'Y'],
            [5.5,530,117,'Y'],
            [3.2,470,-170,'N'],
            [5.2,505,-90,'Y'],
            [1.85,465,120,'Y'],
            [4.8,517,147,'Y'],
            [1.7,430,-100,'Y']
    ]
f2=[
            [1.5,450,220,'N'],
            [4.5,520,-120,'N'],
            [3,490,120,'Y'],
            [5.5,530,117,'Y'],
            [3.2,470,-170,'N'],
            [5.2,505,-90,'Y'],
            [1.85,465,120,'Y'],
            [4.8,517,147,'Y'],
            [1.7,430,-100,'Y']
    ]

def level(ff,idx):
    gain = 0
    selected_feature = 0
    num_of_features = len(ff[0])-1
    entropy_of_set = calc_entropy(ff)
    # print(entropy_of_set)
    keys = calc_unique(ff,idx)
    r = 0
    # print(keys)
    for i in keys:
        p=0
        n=0
        for row in ff:
            if i==row[idx] and row[num_of_features]=='Y':
                p+=1
            elif i==row[idx]  and row[num_of_features]=='N':
                n+=1
        # print(p,n)
        print(((n+p)/(len(f)))*calc_B(p/(p+n)))
        r += ((n+p)/(len(f)))*calc_B(p/(p+n))
        # print(r)

    g = entropy_of_set-r
    return r
    # return g

```

```python

def split_data(ff,p,idx):
    # print(ff)
    for i in range(len(ff)):
        if(ff[i][idx]<p):
            ff[i][idx] = 'a'
        else:
            ff[i][idx] = 'b'
    return ff

def calc_entropy(a):
    l = len(a[0])
    b = []
    for i in a:
        b.append(i[l-1])
    # print(b)
    keys, values = np.unique(b,return_counts=True)
    sum_v = values.sum()
    res = list(map(lambda a: -(a/sum_v)*math.log2(a/sum_v), values))
    sum_v = sum(res)
    # print(sum_v)
    return sum_v

def calc_unique(a,idx):
    keys = []
    for i in a:
        if i[idx] not in keys:
            keys.append(i[idx])
    return keys

def calc_B(q):
    if q==0:
        return -(1-q)*(math.log2(1-q))
    elif q==1:
        return -q*math.log2(q)
    return -(q*math.log2(q)+(1-q)*(math.log2(1-q)))
for i in range(3):
    g = []
    # qwe = f
    for j in f:
        split_point = j[i]
        data  = split_data(f,split_point,i)
        # print(f)
        # print(data)
        g.append(level(data,i))
        for k in range(len(f2)):
            f[k][i] = f2[k][i]

    print(g)
    print('==================================================================')
```

LEVEL 1

Entropy(Parent) for the above dataset : Total number of YES = 6

Total Number of NO = 3

Entropy (Parent) = (-6/9)* log2(6/9) - (3/9)*log2(3/9)

= 0.9182958340544896

Using this entropy, we calculate the Gain Values for this dataset.

Horizontal Angle

1.5

	Yes	No	Remainder	Gain
<1.5	0	0	0.9182958340544896	0.0
>=1.5	6	3		
<=1.5	0	1	0.7211361106303402	0.19715972342414934
>1.5	6	2		

4.5

	Yes	No	Remainder	Gain
<4.5	3	2	0.8999850522344305	0.018310781820059074
>=4.5	3	1		
<=4.5	3	3	0.6666666666666666	0.2516291673878229

>4.5	3	0		

3

	Yes	No	Remainder	Gain
<3	2	1	0.9182958340544896	0.0
>=3	4	2		
<=3	3	1	0.8999850522344305	0.018310781820059074
>3	3	2		

5.5

	Yes	No	Remainder	Gain
<5.5	5	3	0.8483857803777466	0.06991005367674297
>=5.5	1	0		
<=5.5	6	3	0.9182958340544896	0.0
>5.5	0	0		

3.2

	Yes	No	Remainder	Gain
<3.2	3	1	0.8999850522344305	0.018310781820059074
>=3.2	3	2		
<=3.2	3	2	0.8999850522344305	0.018310781820059074
>3.2	3	1		

5.2

	Yes	No	Remainder	Gain
<5.2	4	3	0.7662885502488623	0.15200728380562722
>=5.2	2	0		
<=5.2	5	3	0.8483857803777466	0.06991005367674297
>5.2	1	0		

1.85

	Yes	No	Remainder	Gain
<1.85	1	1	0.8935382199962686	0.024757614058220967
>=1.85	5	2		
<=1.85	2	1	0.9182958340544896	0.0
>1.85	4	2		

4.8

	Yes	No	Remainder	Gain
<4.8	3	3	0.6666666666666666	0.2516291673878229
>=4.8	3	0		
<=4.8	4	3	0.7662885502488623	0.15200728380562722
>4.8	2	0		

1.7

	Yes	No	Remainder	Gain
<1.7	0	1	0.7211361106303402	0.19715972342414934

>=1.7	6	2		
<=1.7	1	1	0.8935382199962686	0.024757614058220967
>1.7	5	2		

Best Gain for Horizontal Angle : 0.2516291673878229 for the horizontal angle 4.8.

Entropy for the same : 0.6666666666666666

Distance

450

	Yes	No	Remainder	Gain
<450	1	0	0.8483857803777466	0.06991005367674297
>=450	5	3		
<=450	1	1	0.8935382199962686	0.024757614058220967
>450	5	2		

520

	Yes	No	Remainder	Gain
<520	5	2	0.8935382199962686	0.024757614058220967
>=520	1	1		
<=520	5	3	0.8483857803777466	0.06991005367674297
>520	1	0		

490

	Yes	No	Remainder	Gain
<490	2	2	0.8455156082707569	0.07278022578373267
>=490	4	0		
<=490	3	2	0.8999850522344305	0.018310781820059074
>490	3	1		

530

	Yes	No	Remainder	Gain
<530	5	3	0.8483857803777466	0.06991005367674297
>=530	1	0		
<=530	6	3	0.9182958340544896	0.0
>530	0	0		

470

	Yes	No	Remainder	Gain
<470	2	1	0.9182958340544896	0.0
>=470	4	2		
<=470	2	2	0.8455156082707569	0.07278022578373267
>470	4	1		

505

	Yes	No	Remainder	Gain

<505	3	2	0.8999850522344305	0.018310781820059074
>=505	3	1		
<=505	4	2	0.9182958340544896	0.0
>505	2	1		

465

	Yes	No	Remainder	Gain
<465	1	1	0.8935382199962686	0.024757614058220967
>=465	5	2		
<=465	2	1	0.9182958340544896	0.0
>465	4	2		

517

	Yes	No	Remainder	Gain
<517	4	2	0.9182958340544896	0.0
>=517	2	1		
<=517	5	2	0.8935382199962686	0.024757614058220967
>517	1	1		

430

	Yes	No	Remainder	Gain
<430	0	0	0.9182958340544896	0.0
>=430	6	3		

<=430	1	0	0.8483857803777466	0.06991005367674297
>430	5	3		

Wind Speed

220

	Yes	No	Remainder	Gain
<220	6	2	0.7211361106303402	0.19715972342414934
>=220	0	1		
<=220	6	3	0.9182958340544896	0.0
>220	0	0		

-120

	Yes	No	Remainder	Gain
<-120	0	1	0.7211361106303402	0.19715972342414934
>=-120	6	2		
<=-120	0	2	0.4601899388973658	0.45810589515712374
>-120	6	1		

120

	Yes	No	Remainder	Gain
<120	3	2	0.8999850522344305	0.018310781820059074
>=120	3	1		
<=120	5	2	0.8935382199962686	0.024757614058220967
>120	1	1		

117

	Yes	No	Remainder	Gain
<117	2	2	0.8455156082707569	0.07278022578373267
>=117	4	1		
<=117	3	2	0.8999850522344305	0.018310781820059074
>117	3	1		

-170

	Yes	No	Remainder	Gain
<-170	0	0	0.9182958340544896	0.0
>=-170	6	3		
<=-170	0	1	0.7211361106303402	0.19715972342414934
>-170	6	2		

-90

	Yes	No	Remainder	Gain
<-90	1	2	0.7394468924503992	0.17884894160409037

>=-90	5	1		
<=-90	2	2	0.8455156082707569	0.07278022578373267
>-90	4	1		

120

	Yes	**No**	**Remainder**	**Gain**
<120	3	2	0.8999850522344305	0.018310781820059074
>=120	3	1		
<=120	5	2	0.8935382199962686	0.024757614058220967
>120	1	1		

147

	Yes	**No**	**Remainder**	**Gain**
<147	5	2	0.8935382199962686	0.024757614058220967
>=147	1	1		
<=147	6	2	0.7211361106303402	0.19715972342414934
>147	0	1		

-100

	Yes	**No**	**Remainder**	**Gain**
<-100	0	2	0.4601899388973658	0.45810589515712374
>=-100	6	1		

<=-100	1	2	0.7394468924503992	0.17884894160409037
>-100	5	1		

Attribute to be chosen for the root node : ‘Wind Speed’ with the split point=-100 i.e. < -100 >=-100 will be the 2 branches from the root node.

Calculations :

<-100 -> Yes : 0, No : 2, Entropy for this child = 0

>= -100 -> Yes : 6, No : 1 , Entropy for this child = (-6/9 log2(6/9)) - (1/9)log2(1/9)

= 0.5916727785823275

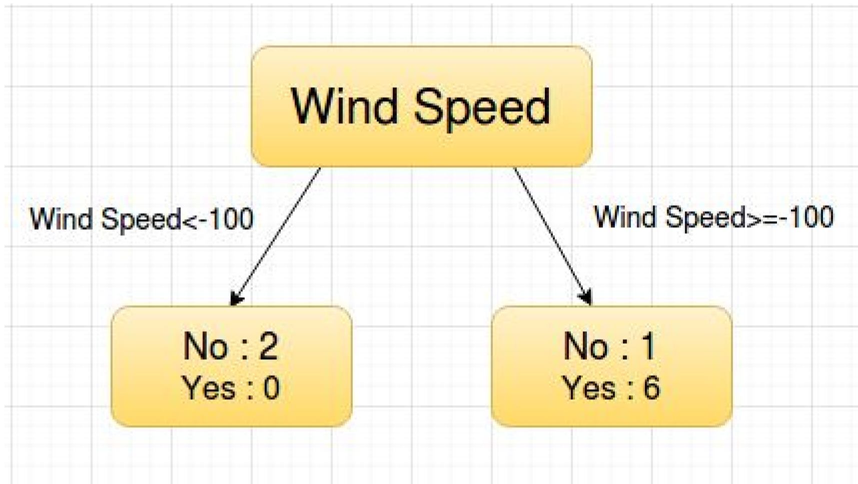
Remainder = ((0+2)/(6+3)) * (0) + ((6+1)/(6+3)) * (0.5916727785823275)

= 0.46018993388973658

Gain = 0.9182958340544896 – 0.46018993388973658

= 0.45810589515712374

This split point is selected because it gives **maximum gain value ~0.4581**



The left child of the Root Node becomes a pure leaf node as it has only 2 No values and 0 Yes values. Thus we will not go beyond that leaf node and will choose an internal node for level 2 for the right hand child as it is still impure.

LEVEL 2 :

New data set:

Horizontal Angle(degree)	Distance(m)	Wind Speed(mph)	Kill
1.5	450	220	N
3	490	120	Y

5.5	530	117	Y
5.2	505	-90	Y
1.85	465	120	Y
4.8	517	147	Y
1.7	430	-100	Y

Entropy(Parent) for the above dataset : Total number of YES = 6

Total Number of NO = 1

Entropy(Parent) = (-6/7)* log2(6/7) - (1/7)*log2(1/7)

= 0.5916727785823275

Using this entropy, we calculate the Gain Values.

<table>
<tr><td>Horizontal Angle</td><td></td></tr>
<tr><td>1.5</td><td>
<table>
<tr><td></td><td>Yes</td><td>No</td><td>Remainder</td><td>Gain</td></tr>
<tr><td><1.5</td><td>0</td><td>0</td><td>0.5916727785823275</td><td>0.0</td></tr>
<tr><td>>=1.5</td><td>6</td><td>1</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td><=1.5</td><td>0</td><td>1</td><td>0.0</td><td>0.5916727785823275</td></tr>
<tr><td>>1.5</td><td>6</td><td>0</td><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>3</td><td>
<table>
<tr><td></td><td>Yes</td><td>No</td><td>Remainder</td><td>Gain</td></tr>
</table>
</td></tr>
</table>

<3	2	1	0.39355535745192405	0.19811742113040343
>=3	4	0		
<=3	3	1	0.46358749969093305	0.12808527889139443
>3	3	0		

5.5

	Yes	**No**	**Remainder**	**Gain**
<5.5	5	1	0.5571620756985892	0.03451070288373825
>=5.5	1	0		
<=5.5	6	1	0.5916727785823275	0.0
>5.5	0	0		

5.2

	Yes	**No**	**Remainder**	**Gain**
<5.2	4	1	0.5156629249195446	0.0760098536627829
>=5.2	2	0		
<=5.2	5	1	0.5571620756985892	0.03451070288373825
>5.2	1	0		

1.85

	Yes	**No**	**Remainder**	**Gain**
<1.85	1	1	0.2857142857142857	0.3059584928680418
>=1.85	5	0		

<=1.85	2	1	0.39355535745192405	0.19811742113040343
>1.85	4	0		

4.8

	Yes	**No**	**Remainder**	**Gain**
<4.8	3	1	0.46358749969093305	0.12808527889139443
>=4.8	3	0		
<=4.8	4	1	0.5156629249195446	0.0760098536627829
>4.8	2	0		

1.7

	Yes	**No**	**Remainder**	**Gain**
<1.7	0	1	0.0	0.5916727785823275
>=1.7	6	0		
<=1.7	1	1	0.2857142857142857	0.3059584928680418
>1.7	5	0		

<table>
<tr><th>Distance</th><th></th></tr>
<tr><td>450</td><td>
<table>
<tr><th></th><th>Yes</th><th>No</th><th>Remainder</th><th>Gain</th></tr>
<tr><td><450</td><td>1</td><td>0</td><td>0.5571620756985892</td><td>0.03451070288373825</td></tr>
<tr><td>>=450</td><td>5</td><td>1</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td><=450</td><td>1</td><td>1</td><td>0.2857142857142857</td><td>0.3059584928680418</td></tr>
<tr><td>>450</td><td>5</td><td>0</td><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>490</td><td>
<table>
<tr><th></th><th>Yes</th><th>No</th><th>Remainder</th><th>Gain</th></tr>
<tr><td><490</td><td>2</td><td>1</td><td>0.39355535745192405</td><td>0.19811742113040343</td></tr>
<tr><td><=490</td><td>4</td><td>0</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td><=490</td><td>3</td><td>1</td><td>0.46358749969093305</td><td>0.12808527889139443</td></tr>
<tr><td>>490</td><td>3</td><td>0</td><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>530</td><td>
<table>
<tr><th></th><th>Yes</th><th>No</th><th>Remainder</th><th>Gain</th></tr>
<tr><td><530</td><td>5</td><td>1</td><td>0.5571620756985892</td><td>0.03451070288373825</td></tr>
<tr><td>>=530</td><td>1</td><td>0</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td><=530</td><td>6</td><td>1</td><td>0.5916727785823275</td><td>0.0</td></tr>
<tr><td>>530</td><td>0</td><td>0</td><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>505</td><td>
<table>
<tr><th></th><th>Yes</th><th>No</th><th>Remainder</th><th>Gain</th></tr>
</table>
</td></tr>
</table>

<505	3	1	0.46358749969093305	0.12808527889139443
>=505	3	0		
<=505	4	1	0.5156629249195446	0.0760098536627829
>505	2	0		

465

	Yes	**No**	**Remainder**	**Gain**
<465	1	1	0.2857142857142857	0.3059584928680418
>=465	5	0		
<=465	2	1	0.39355535745192405	0.19811742113040343
>465	4	0		

517

	Yes	**No**	**Remainder**	**Gain**
<517	4	1	0.5156629249195446	0.0760098536627829
>=517	2	0		
<=517	5	1	0.5571620756985892	0.034510702883738250
>517	1	0		

430

	Yes	**No**	**Remainder**	**Gain**
<430	0	0	0.5916727785823275	0.0
>=430	6	1		

<=430	1	0	0.5571620756985892	0.03451070288373825
>430	5	1		

Attribute to be chosen for the next internal node : 'Horizontal Angle' with the split point= 1.7 i.e. <1.7 and >=1.7 will be the 2 branches from the 2nd level node

Calculations:

<1.7 -> Yes : 0, No : 1, Entropy for this child = 0

>= 1.7 -> Yes : 6, No : 0 , Entropy for this child = 0

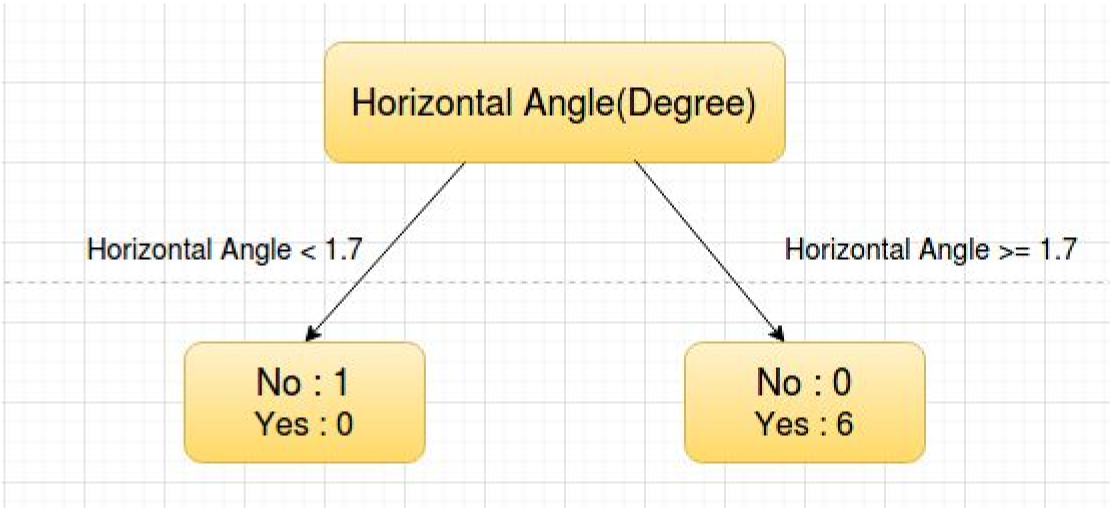
Remainder = ((0+1)/(6+1)) * (0) + ((6+0)/(6+1)) * (0)

= 0.0

Gain = 0.5916727785823275 – 0.0

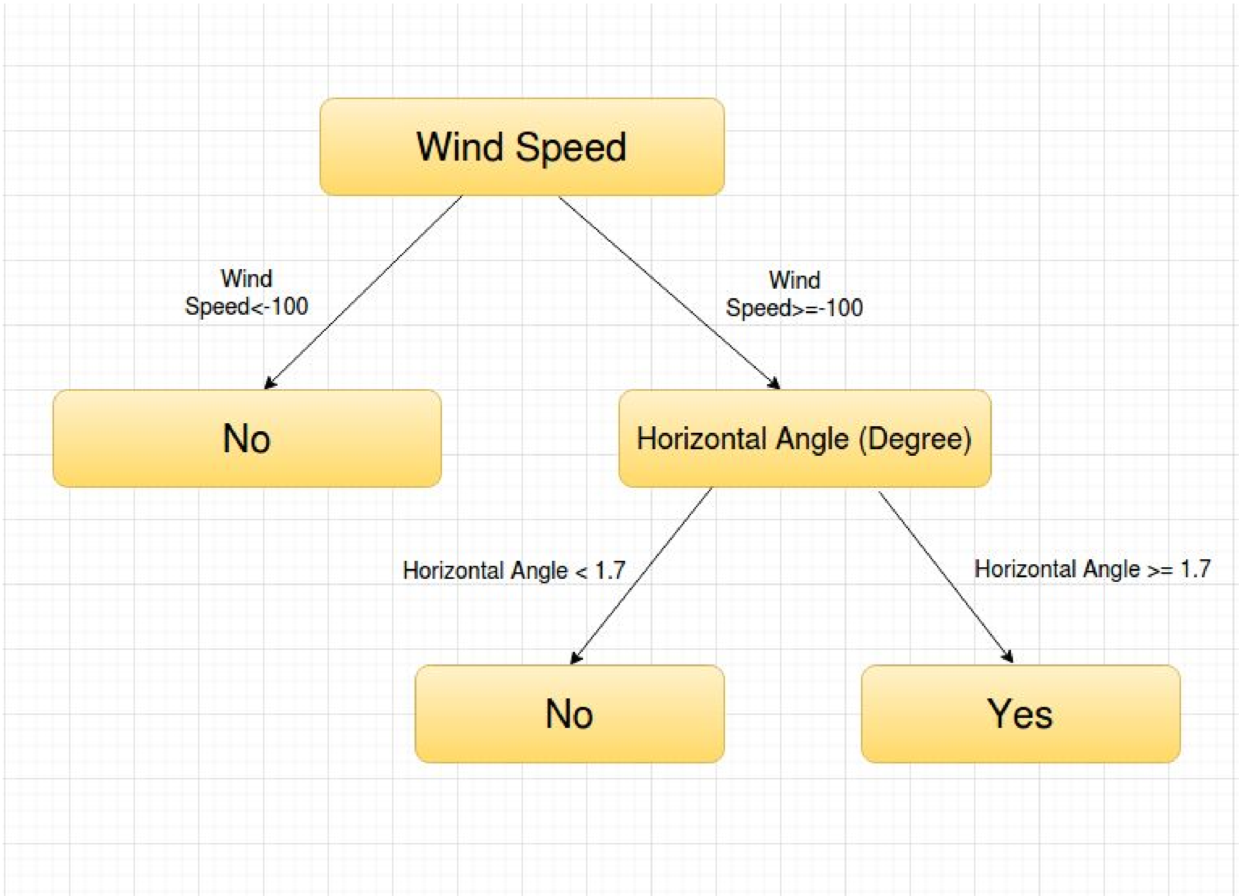
= 0.5916727785823275

This split point is selected because it gives **maximum gain value ~ 0.5916**



Final Decision Tree

Final tree obtained is of height 2 i.e 2 levels.



The left child of the Horizontal Angle Node becomes a pure leaf node as it has only 1 No value and 0 Yes values. Thus we will not go beyond that leaf node.

The right child of the same node also becomes a pure leaf node as it has 0 No values and 6 Yes Values. Thus we will not go beyond that leaf node too.

Since for all nodes, we have reached the leaf nodes, and hence we have the decision tree for the original dataset.