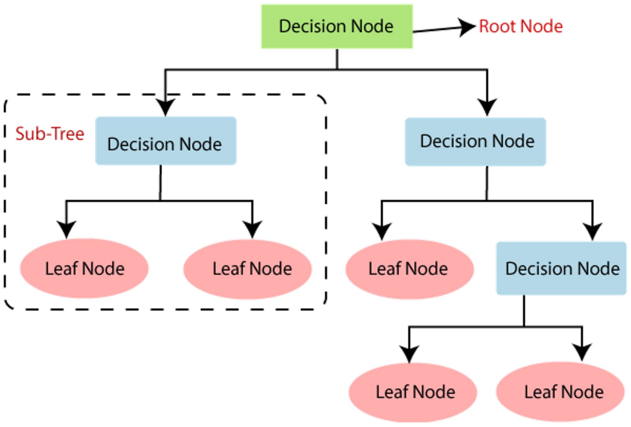
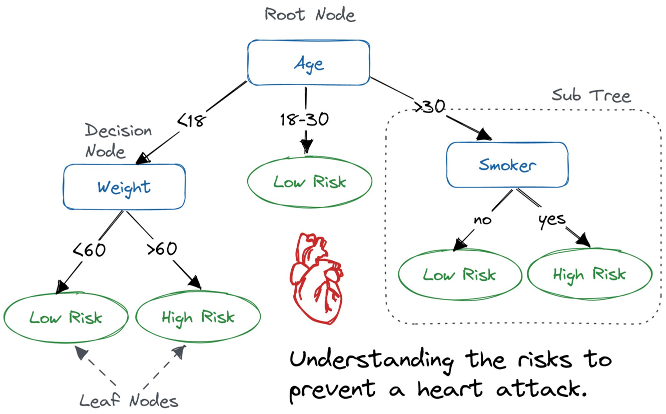
DECISION TREE MACHINE LEARNING ALGORITHM

> Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.





Attribute Selection Measures (DT Optimizers)

> While implementing a Decision tree, the main issue arises that how to select the best attribute for the root node and for sub-nodes. So, to solve such problems there is a technique which is called as Attribute selection measure or ASM. By this measurement, we can easily select the best attribute for the nodes of the tree. There are two popular techniques for ASM, which are:

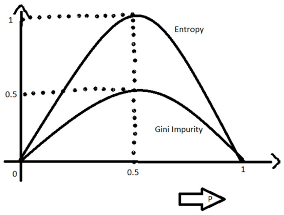
1. Information Gain (Entropy)

> Information gain is the measurement of changes in entropy after the segmentation of a dataset based on an attribute. It calculates how much information a feature provides us about a class. According to the value of information gain, we split the node and build the decision tree. A decision tree algorithm always tries to maximize the value of information gain, and a node/attribute having the highest information gain is split first. Entropy is a logarithmic measure.

- Entropy: Entropy is a metric to measure the impurity in a given attribute. It specifies randomness in data.

2. Gini Index

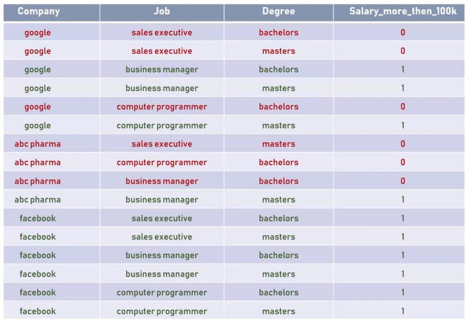
> Gini index is a measure of impurity or purity used while creating a decision tree in the CART (Classification and Regression Tree) algorithm. An attribute with the low Gini index should be preferred as compared to the high Gini index. It only creates binary splits, and the CART algorithm uses the Gini index to create binary splits. Gini index is a linear measure.



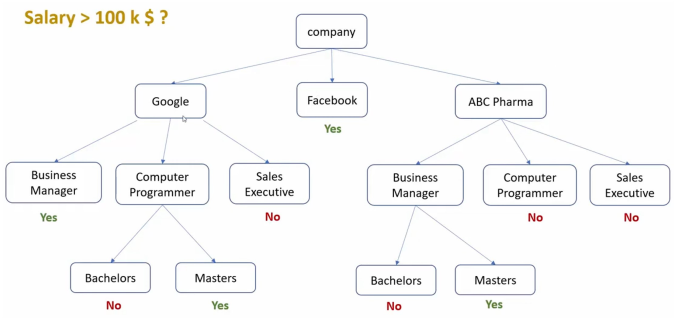
$$E(S) = \sum_{i=1}^{c} -p_i \log_2 p_i$$

$$Gini(E) = 1 - \sum_{j=1}^{c} p_j^2$$

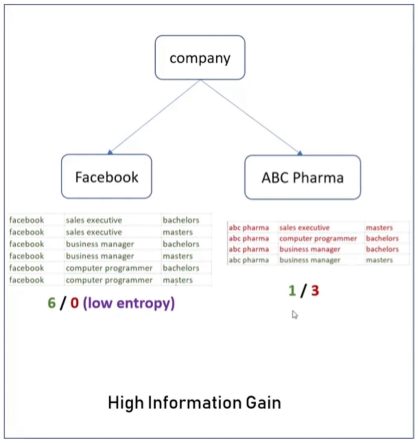
Example Problem:



Company	Job	Degree	Salary_more_then_100k
google	sales executive	bachelors	0
google	sales executive	masters	0
google	business manager	bachelors	1
google	business manager	masters	1
google	computer programmer	bachelors	0
google	computer programmer	masters	1
abc pharma	sales executive	masters	0
abc pharma	computer programmer	bachelors	0
abc pharma	business manager	bachelors	0
abc pharma	business manager	masters	1
facebook	sales executive	bachelors	1
facebook	sales executive	masters	1
facebook	business manager	bachelors	1
facebook	business manager	masters	1
facebook	computer programmer	bachelors	1
facebook	computer programmer	masters	1



Q: How do you select ordering of features ?



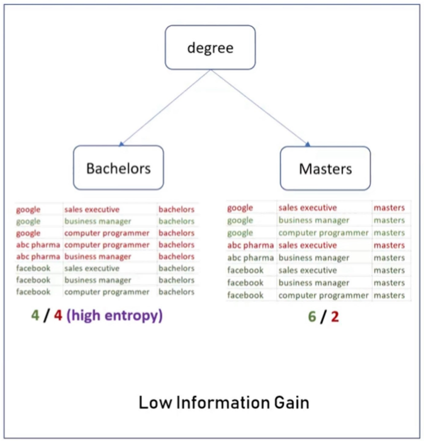
facebook	sales executive	bachelors
facebook	sales executive	masters
facebook	business manager	bachelors
facebook	business manager	masters
facebook	computer programmer	bachelors
facebook	computer programmer	masters

6 / 0 (low entropy)

abc pharma	sales executive	masters
abc pharma	computer programmer	bachelors
abc pharma	business manager	bachelors
abc pharma	business manager	masters

1 / 3

High Information Gain



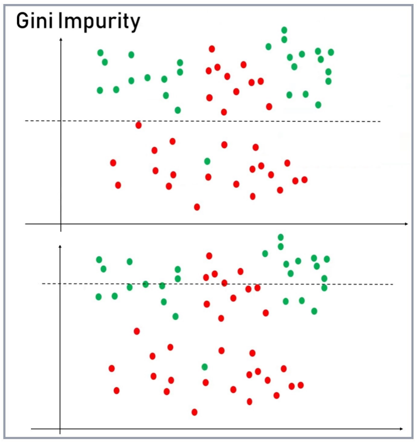
google	sales executive	bachelors
google	business manager	bachelors
google	computer programmer	bachelors
abc pharma	computer programmer	bachelors
abc pharma	business manager	bachelors
facebook	sales executive	bachelors
facebook	business manager	bachelors
facebook	computer programmer	bachelors

4 / 4 (high entropy)

google	sales executive	masters
google	business manager	masters
google	computer programmer	masters
abc pharma	sales executive	masters
abc pharma	business manager	masters
facebook	sales executive	masters
facebook	business manager	masters
facebook	computer programmer	masters

6 / 2

Low Information Gain



In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
from matplotlib import pyplot as plt
from sklearn.preprocessing import LabelEncoder
from sklearn.tree import DecisionTreeClassifier
```

In [2]:
```
df = pd.read_csv("salaries.csv")
df
```

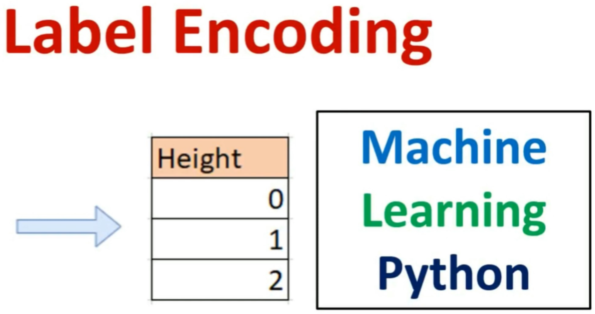
Out[2]:

	company	job	degree	salary_more_then_100k
0	google	sales executive	bachelors	0

1	google	sales executive	masters	0
2	google	business manager	bachelors	1
3	google	business manager	masters	1
4	google	computer programmer	bachelors	0
5	google	computer programmer	masters	1
6	abc pharma	sales executive	masters	0
7	abc pharma	computer programmer	bachelors	0
8	abc pharma	business manager	bachelors	0
9	abc pharma	business manager	masters	1
10	facebook	sales executive	bachelors	1
11	facebook	sales executive	masters	1
12	facebook	business manager	bachelors	1
13	facebook	business manager	masters	1
14	facebook	computer programmer	bachelors	1
15	facebook	computer programmer	masters	1

In [3]:

```
x = df.drop('salary_more_then_100k',axis='columns')
y = df['salary_more_then_100k']
```



Height
Tall
Medium
Short

Height
0
1
2

In [4]:

```
le_company = LabelEncoder()
le_job = LabelEncoder()
le_degree = LabelEncoder()
x['company_n']  =   le_company.fit_transform(x['company'])
x['job_n']      =   le_job.fit_transform(x['job'])
x['degree_n']   =   le_degree.fit_transform(x['degree'])
```

In [5]:

```
x
```

Out[5]:

	company	job	degree	company_n	job_n	degree_n
0	google	sales executive	bachelors	2	2	0
1	google	sales executive	masters	2	2	1
2	google	business manager	bachelors	2	0	0
3	google	business manager	masters	2	0	1
4	google	computer programmer	bachelors	2	1	0
5	google	computer programmer	masters	2	1	1
6	abc pharma	sales executive	masters	0	2	1

7	abc pharma	computer programmer	bachelors	0	1	0
8	abc pharma	business manager	bachelors	0	0	0
9	abc pharma	business manager	masters	0	0	1
10	facebook	sales executive	bachelors	1	2	0
11	facebook	sales executive	masters	1	2	1
12	facebook	business manager	bachelors	1	0	0
13	facebook	business manager	masters	1	0	1
14	facebook	computer programmer	bachelors	1	1	0
15	facebook	computer programmer	masters	1	1	1

In [6]:

```
X = x.drop(['company','job','degree'],axis='columns')
X
```

Out[6]:

	company_n	job_n	degree_n
0	2	2	0
1	2	2	1
2	2	0	0
3	2	0	1
4	2	1	0
5	2	1	1
6	0	2	1
7	0	1	0
8	0	0	0
9	0	0	1
10	1	2	0
11	1	2	1
12	1	0	0
13	1	0	1
14	1	1	0
15	1	1	1

In [7]:

```
y.values
```

Out[7]:

```
array([0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1], dtype=int64)
```

In [8]:

```
model = DecisionTreeClassifier()
model.fit(X.values, y)
model.score(X.values, y)
```

Out[8]:

```
1.0
```

Is salary of Google, Computer Engineer, Bachelors degree > 100 k ?

In [9]:

```
p1 = model.predict([[2,1,0]])
print(str(p1) + ' = No')
```

```
[0] = No
```

Is salary of Google, Computer Engineer, Masters degree > 100 k ?

In [10]:

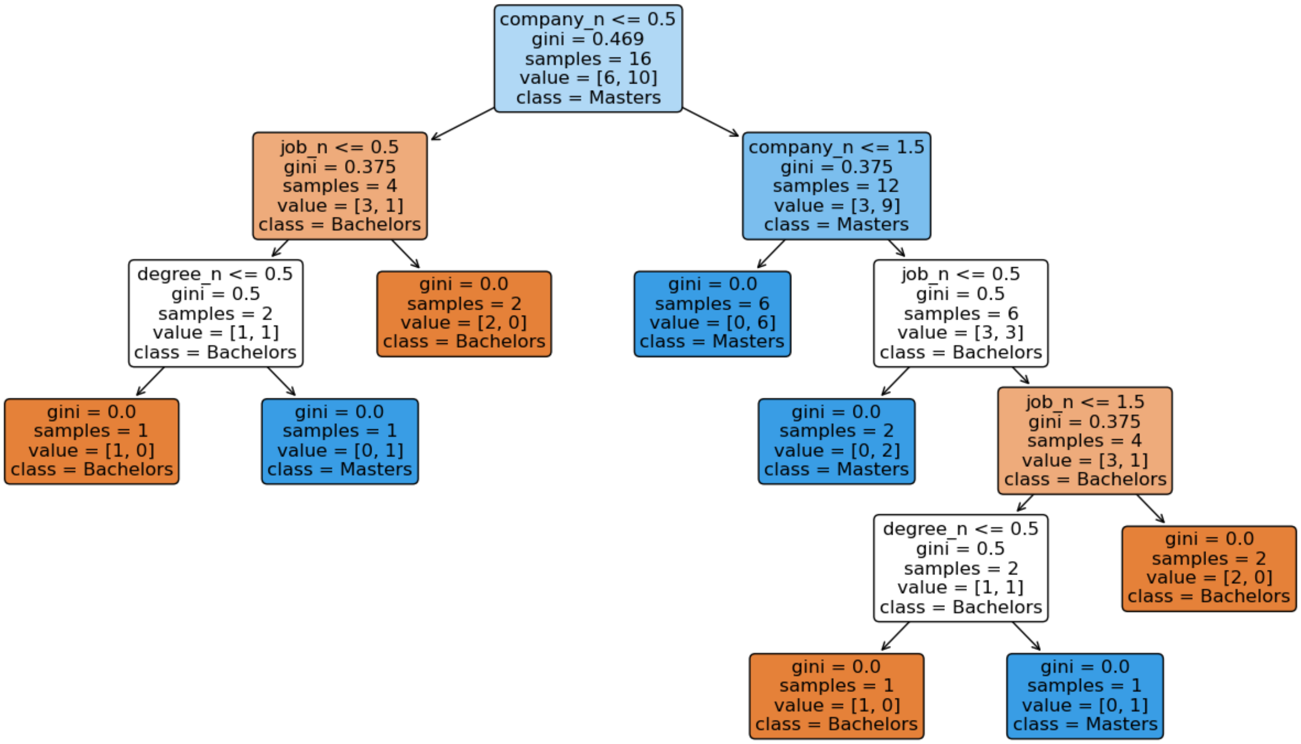
```
p2 = model.predict([[2,1,1]])
print(str(p2) + ' = Yes')
```

```
[1] = Yes
```

In [11]:

```
from sklearn import tree
from matplotlib import pyplot as plt

fig = plt.figure(figsize=(16,9))
fig  = tree.plot_tree(model,
                      feature_names=X.columns,
                      class_names={0:'Bachelors', 1:'Masters'},
                      filled=True,
                      rounded=True,
                      fontsize=12)
```


company_n <= 0.5
gini = 0.469
samples = 16
value = [6, 10]
class = Masters
job_n <= 0.5
gini = 0.375
samples = 4
value = [3, 1]
class = Bachelors
company_n <= 1.5
gini = 0.375
samples = 12
value = [3, 9]
class = Masters
degree_n <= 0.5
gini = 0.5
samples = 2
value = [1, 1]
class = Bachelors
gini = 0.0
samples = 2
value = [2, 0]
class = Bachelors
gini = 0.0
samples = 6
value = [0, 6]
class = Masters
job_n <= 0.5
gini = 0.5
samples = 6
value = [3, 3]
class = Bachelors
gini = 0.0
samples = 1
value = [1, 0]
class = Bachelors
gini = 0.0
samples = 1
value = [0, 1]
class = Masters
gini = 0.0
samples = 2
value = [0, 2]
class = Masters
job_n <= 1.5
gini = 0.375
samples = 4
value = [3, 1]
class = Bachelors
degree_n <= 0.5
gini = 0.5
samples = 2
value = [1, 1]
class = Bachelors
gini = 0.0
samples = 2
value = [2, 0]
class = Bachelors
gini = 0.0
samples = 1
value = [1, 0]
class = Bachelors
gini = 0.0
samples = 1
value = [0, 1]
class = Masters