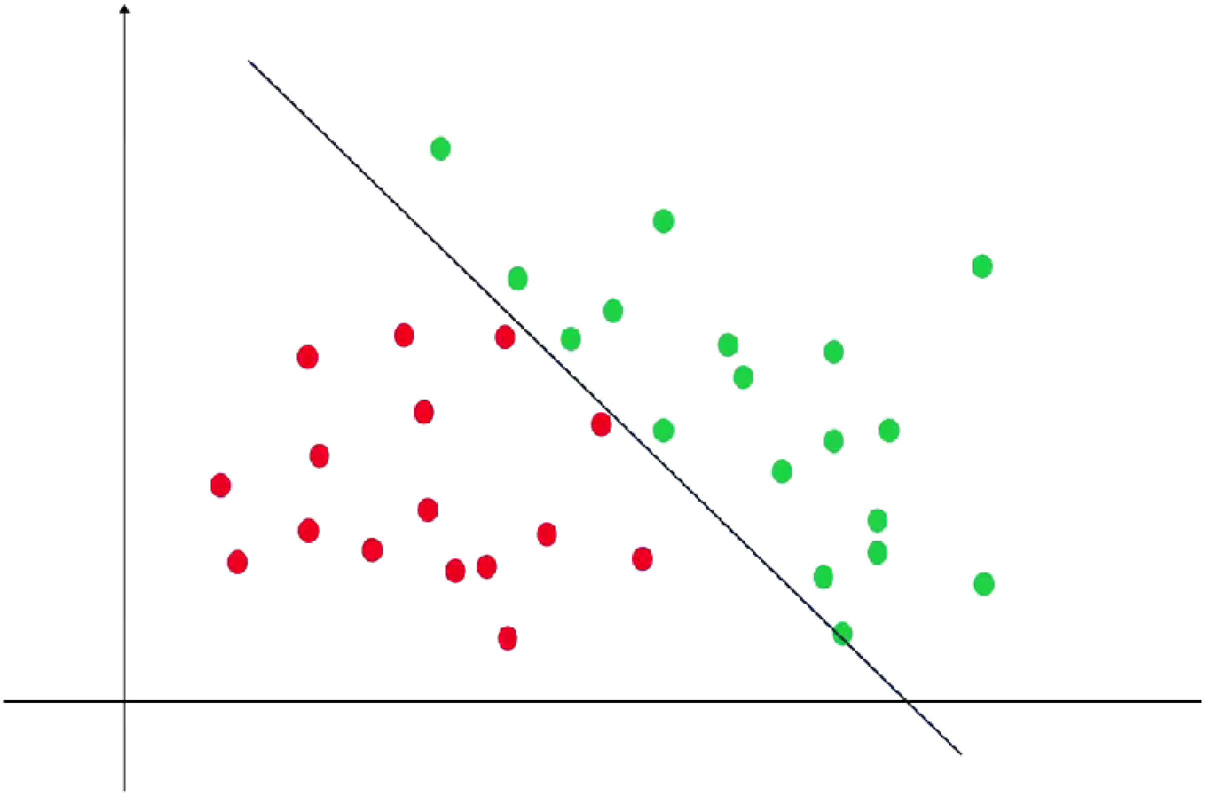
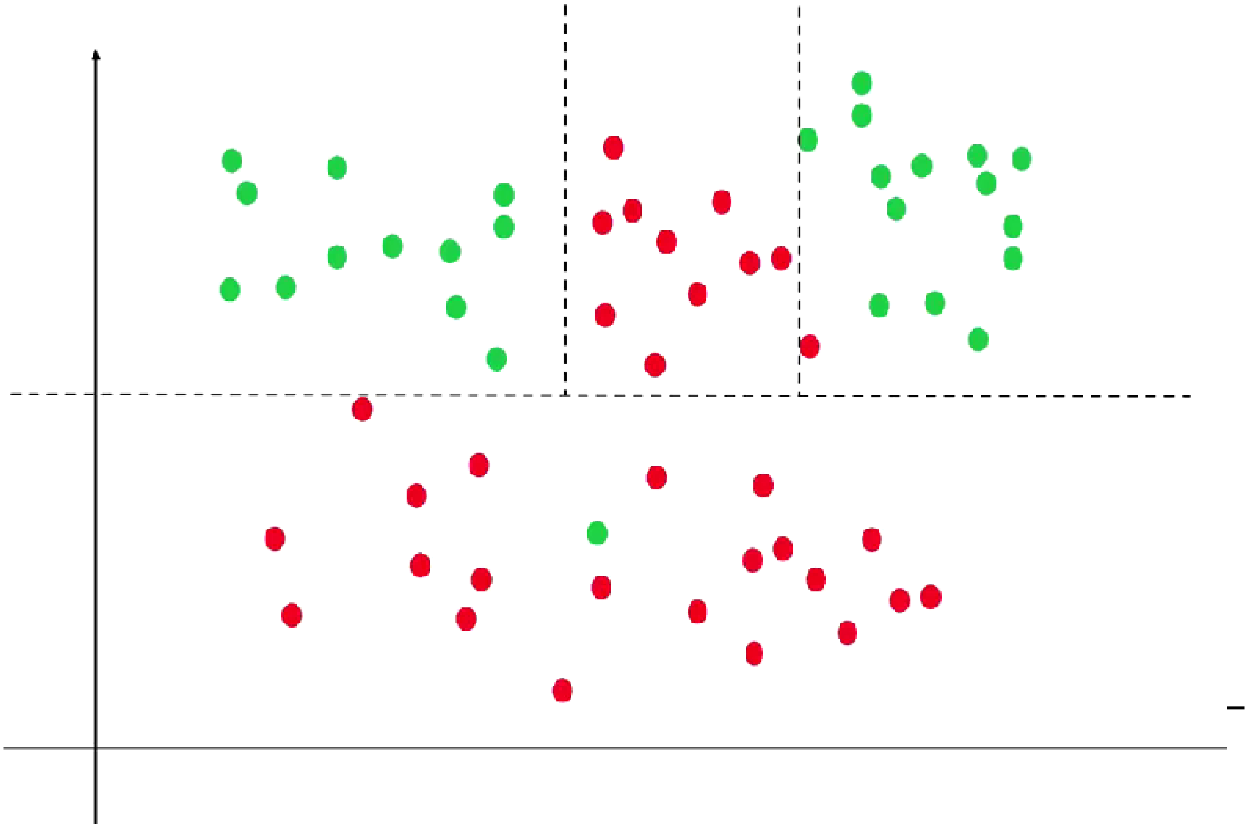
Classification Problem:Decision Tree

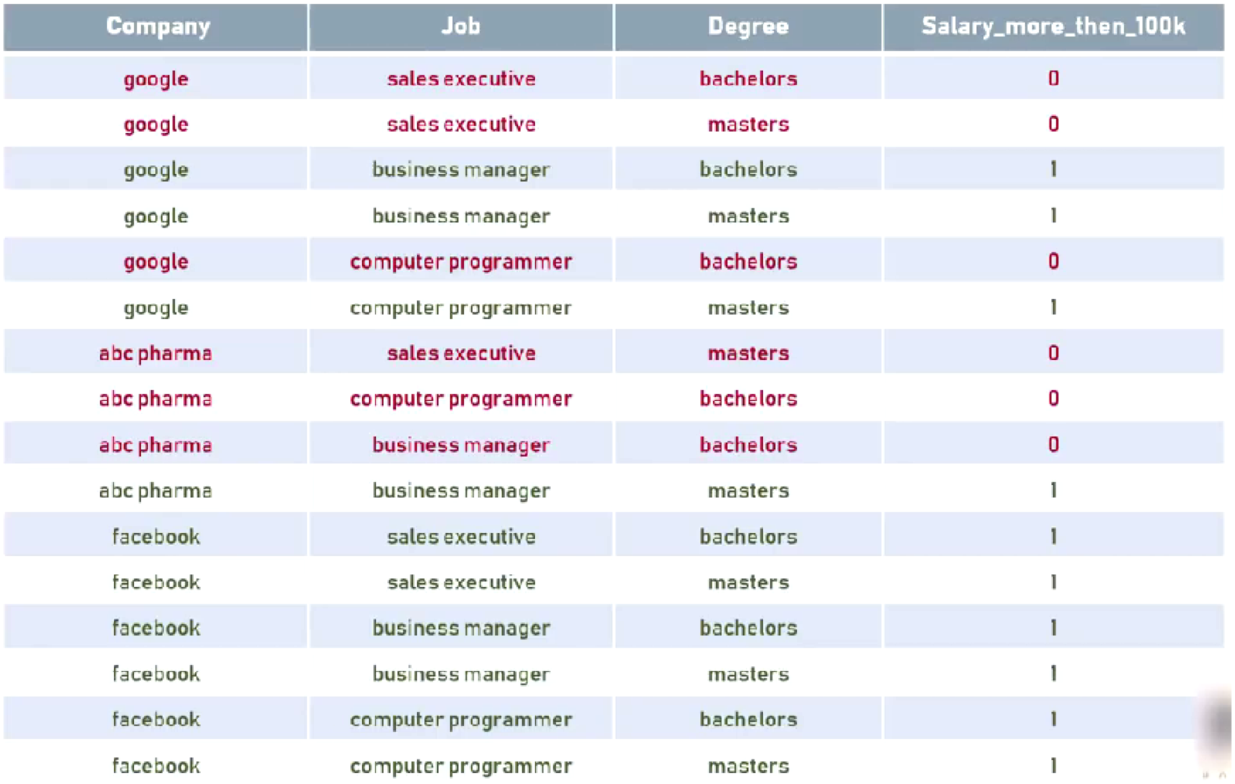
When you have a dataset like this it's easier to draw a decision boundry using LogisticRegression.



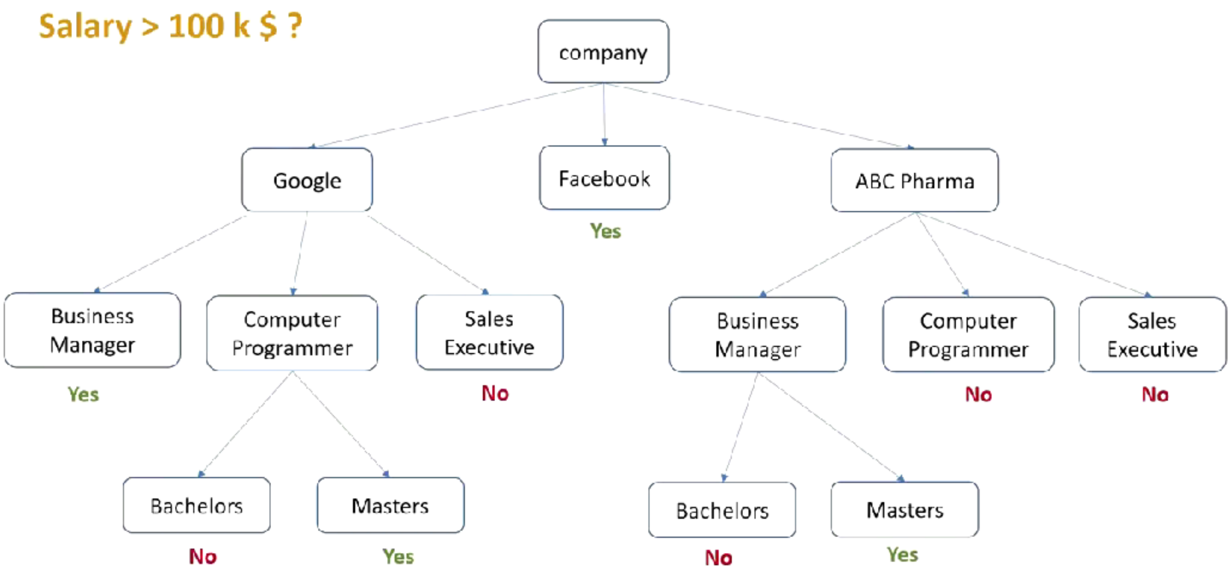
But if your dataset is complicate like this you cannot just draw a single line. You might have to split your dataset again and again to come up with the decision boundaries and this is what decision algorithm does for you.



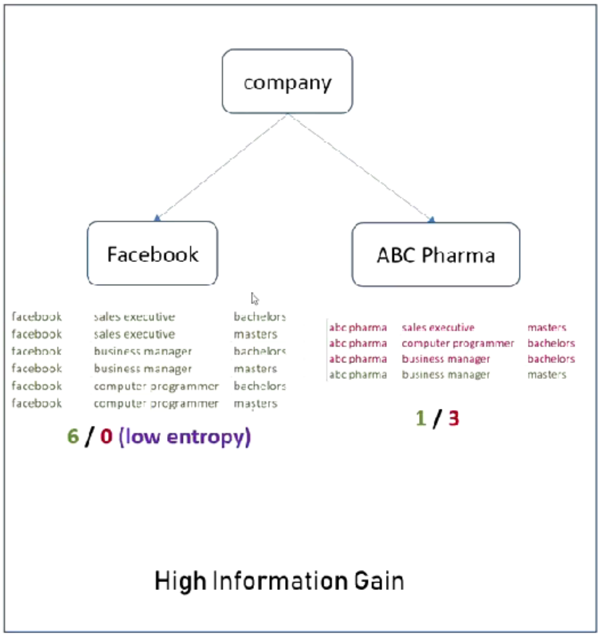
Problem: We will use this particular dataset where you try to predict if person salaries more than 100K$ based on the company and the job title and the degree that he has.



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



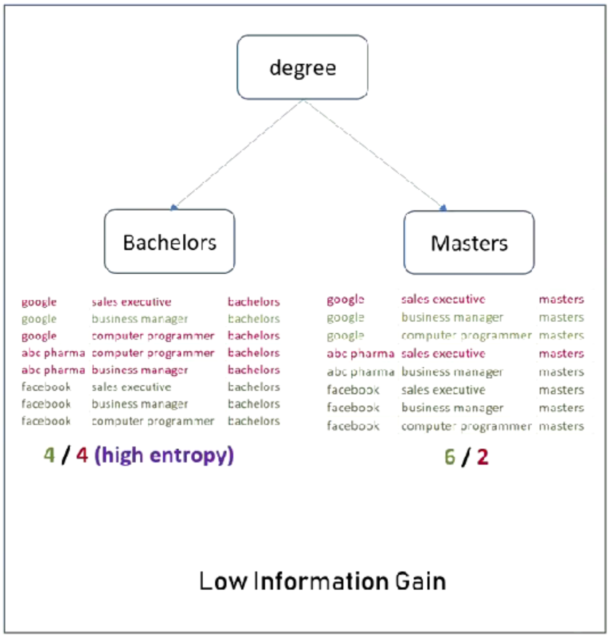
How do you select ordering of features?



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

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

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

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

Low Information Gain

entropy is the measure of randomness in your samples. low entropy means there is no randomness. low entropy => High information gain => better choice for ordering features.

In [1]:
```
import pandas as pd
```

In [3]:
```
df = pd.read_csv("D:/Data_Science/My Github/Machine-Learning-with-Python/9. Decisi
df.head()
```

Out[3]:

	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

In [4]:
```
# divided my dataframe between independant variable(inputs) & target varibable
inputs = df.drop('salary_more_then_100k',axis='columns')
```

In [5]:
```
target = df['salary_more_then_100k']
```

ML algorithm can only work on numbers, so we have to converts input columns into numbers. One way to do this in to use LabelEncoder.

In [6]:
```
from sklearn.preprocessing import LabelEncoder
```

I have 3 columns so I have to create 3 different objects.

In [7]:
```
le_company = LabelEncoder()
le_job = LabelEncoder()
le_degree = LabelEncoder()
```

Once you have those 3, in your inputs dataframe you are creating 1 more column for each by using fit_transform method to convert them into numbers.

In [8]:
```
inputs['company_n'] = le_company.fit_transform(inputs['company'])
inputs['job_n'] = le_job.fit_transform(inputs['job'])
inputs['degree_n'] = le_degree.fit_transform(inputs['degree'])
```

In [9]:

```
inputs
```

Out[9]:

	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

Next step is to drop the real columns for just having numbers.

In [10]:

```
inputs_n = inputs.drop(['company','job','degree'],axis='columns')
```

In [16]:

```
inputs_n
```

Out[16]:

	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 [17]:

```
target
```

Out[17]:

```
0     0
1     0
2     1
3     1
4     0
5     1
6     0
7     0
8     0
9     1
10    1
11    1
12    1
13    1
14    1
15    1
Name: salary_more_then_100k, dtype: int64
```

train_test_split

In [19]:

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(inputs_n,target,test_size=0.2)
```

```
In [20]: len(X_train)

Out[20]: 12
```

```
In [21]: len(X_test)

Out[21]: 4
```

Now we are ready to train our classifier.

```
In [22]: from sklearn import tree
         model = tree.DecisionTreeClassifier()
```

```
In [24]: model.fit(X_train,y_train)

Out[24]: DecisionTreeClassifier()
```

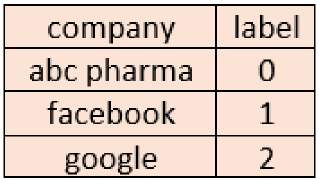
```
In [25]: model.score(X_test,y_test)

Out[25]: 0.5
```

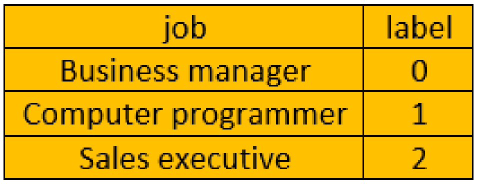
This is because the number of data is too small.

Prediction

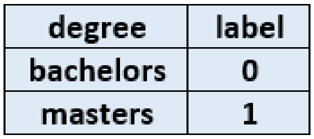
Is salary of Google, Sales executive, Masters degree > 100 k ?



company	label
abc pharma	0
facebook	1
google	2



job	label
Business manager	0
Computer programmer	1
Sales executive	2



degree	label
bachelors	0
masters	1

```
In [27]: model.predict([[2,2,1]])

Out[27]: array([0], dtype=int64)
```

It means this person's salary is not greater than 100K.

Is salary of facebook, business manager, bachelor degree > 100 k ?

```
In [28]: model.predict([[1,0,0]])

Out[28]: array([1], dtype=int64)
```

It means that this persron's salary > 100K.

Date	Author
2021-09-13	**Ehsan Zia**