Group Assignment 4

Group 7
Prof. Dr Dozier
Armin Khayyer, Bhargav Joshi, Ye Wang

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

dictinary = {"Day":["D1", "D2", "D3", "D4", "D5", "D6", "D7", "D8", "D9", "D10", "D11", "D12", "D13", "D14"],
             "OUTLOOK": ["Sunny", "Sunny", "Overcast", "Rain", "Rain", "Rain", "Overcast", "Sunny","Sunny",
                         "Rain", "Sunny", "Overcast", "Overcast", "Rain"],
             "Temperature":["Hot", "Hot", "Hot", "Mild", "Cool", "Cool", "Mild", "Cool",
                            "Mild", "Mild", "Mild", "Hot", "Mild"],
             "Humidity": ["High","High","High","High", "Normal","Normal","Normal","High","Normal",
                          "Normal","Normal","High","Normal", "High" ],
             "Wind":["Weak", "Strong","Weak","Weak","Weak","Strong","Strong", "Weak","Weak","Weak",
                     "Strong","Strong","Weak","Strong"],
             "PlayTennis":["No", "No", "Yes", "Yes", "Yes", "No", "Yes","No", "Yes", "Yes", "Yes",
                           "Yes", "Yes", "No"]}

df = pd.DataFrame(dictinary)
df
```

Out[1]:

	Day	OUTLOOK	Temperature	Humidity	Wind	PlayTennis
0	D1	Sunny	Hot	High	Weak	No
1	D2	Sunny	Hot	High	Strong	No
2	D3	Overcast	Hot	High	Weak	Yes
3	D4	Rain	Mild	High	Weak	Yes
4	D5	Rain	Cool	Normal	Weak	Yes
5	D6	Rain	Cool	Normal	Strong	No
6	D7	Overcast	Cool	Normal	Strong	Yes
7	D8	Sunny	Mild	High	Weak	No
8	D9	Sunny	Cool	Normal	Weak	Yes
9	D10	Rain	Mild	Normal	Weak	Yes
10	D11	Sunny	Mild	Normal	Strong	Yes
11	D12	Overcast	Mild	High	Strong	Yes
12	D13	Overcast	Hot	Normal	Weak	Yes
13	D14	Rain	Mild	High	Strong	No

In [2]:
```
part1 = len(df.loc[(df.PlayTennis=="Yes")&(df.Humidity == "High"), :])
part2 = len(df.loc[(df.PlayTennis=="Yes")&(df.Humidity == "Normal"), :])

len_df = len(df)
def frac(a, b):
    return r'$\frac{'+str(a)+'}{'+str(b)+'}$'
```

Question 1

$\sum_i p(PlayTennis = Yes \cap Humidity_i) = p(PlayTennis = Yes \cap Humidity = High) + p(PlayTennis = Yes \cap Humidity = Normal)$

$= \frac{3}{14} + \frac{6}{14} = \frac{9}{14}$

In [3]:
```
tennis_given_sun_norm = len(df.loc[(df.PlayTennis=="Yes")&(df.Humidity == "Normal") &(df.OUTLOOK == "Sunny"), :])
total_given_sun_normal = len(df.loc[(df.Humidity == "Normal") &(df.OUTLOOK == "Sunny"), :])
```

Question 2

- p(PlayTennis = Yes | Outlook = Sunny, Humidity = Normal) = $\frac{2}{2}$ = 1
- $\frac{p(Outlook=Sunny,Humidity=Normal \quad | \quad PlayTennis=Yes) \quad \times \quad p(PlayTennis=Yes)}{p(Outlook=Sunny,Humidity=Normal)} = \frac{\frac{2}{9}\times\frac{9}{14}}{\frac{2}{14}} = \frac{2}{2} = 1$

This means that the probability of playing Tennis given the outlook is sunny and the humidity is normal is equal to one. Simply becasue in all the data rows that the outlook is sunny and the humidity is normal the player has played Tennis.

In [4]:
```
#yes
Result_yes1 = len(df.loc[(df.OUTLOOK=="Sunny")&(df.PlayTennis == "Yes"), :])
Result_yes2 = len(df.loc[(df.Humidity=="Normal")&(df.PlayTennis == "Yes"), :])
Result_yes3 = len(df.loc[(df.PlayTennis=="Yes"), :])

#no
Result_no1 = len(df.loc[(df.OUTLOOK=="Sunny")&(df.PlayTennis == "No"), :])
Result_no2 = len(df.loc[(df.Humidity=="Normal")&(df.PlayTennis == "No"), :])
Result_no3 = len(df.loc[(df.PlayTennis=="No"), :])
```

Question 3

- Classify: (Outlook = Sunny, Humidity = Normal)
- $Result_{yes} = \frac{p(Outlook=Sunny \quad \cap \quad PlayTennis=Yes)}{p(PlayTennis=Yes)} \times \frac{p(Humidity=Normal \quad \cap \quad PlayTennis=Yes)}{p(PlayTennis=Yes)} \times p(PlayTennis = Yes)$

$$Result_{yes} = \frac{\frac{2}{14}}{\frac{9}{14}} \times \frac{\frac{6}{14}}{\frac{9}{14}} \times \frac{9}{14} = \frac{2}{9} \times \frac{6}{9} \times \frac{9}{14} = \frac{12}{126} = \frac{2}{21}$$

- $Result_{no} = \frac{p(Outlook=Sunny \quad \cap \quad PlayTennis=No)}{p(PlayTennis=No)} \times \frac{p(Humidity=Normal \quad \cap \quad PlayTennis=No)}{p(PlayTennis=No)} \times p(PlayTennis = No)$

$$Result_{No} = \frac{\frac{3}{14}}{\frac{5}{14}} \times \frac{\frac{1}{14}}{\frac{5}{14}} \times \frac{5}{14} = \frac{3}{5} \times \frac{1}{5} \times \frac{5}{14} = \frac{3}{70}$$

$Result_{Yes}$ is higher than $Result_{No}$, therefore the Naive Baysian classifier(NBC) classifies this combination as a "Yes"

Question 4

Naïve Bayesian Classifiers (Continuous)

Day	Outlook	Temperature	Humidity	Wind	PlayTime
D1	Sunny	85	85	Weak	5
D2	Sunny	80	90	Strong	0
D3	Overcast	83	86	Weak	55
D4	Rain	70	96	Weak	40
D5	Rain	68	80	Weak	65
D6	Rain	65	70	Strong	7
D7	Overcast	64	65	Strong	60
D8	Sunny	72	95	Weak	0
D9	Sunny	69	70	Weak	70
D10	Rain	75	80	Weak	45
D11	Sunny	75	70	Strong	50
D12	Overcast	72	90	Strong	55
D13	Overcast	81	75	Weak	75
D14	Rain	71	91	Strong	10

- How would we develop an NBC for this problem?

There are many ways to perform naive Bayes classification (NBC). A common technique in NBC is to recode the feature (variable) values into quartiles, such that values less than the 25th percentile are assigned a 1, 25th to 50th a 2, 50th to 75th a 3 and greater than the 75th percentile a 4. Thus a single object will deposit one count in bin Q1, Q2, Q3, or Q4. Calculations are merely done on these categorical bins. Bin counts (probabilities) are then based on the number of samples whose variable values fall within a given bin. For example, if a set of objects have very high values for feature X1, then this will result in a lot of bin counts in the bin for Q4 of X1. On the other hand, if another set of objects has low values for feature X1, then those objects will deposit a lot of counts in the bin for Q1 of feature X1.

Another approach is to use an unsupervised machine learning methodology such as Kohonen clustering to cluster each feature into a user-specified number of clusters and then use the clusters as discrete features to calculate the probabilities of NBC.

In [ ]: