```
# This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/) that gets preserved as output when you create a ve
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the current session
```

```
/kaggle/input/play-tennis/play_tennis.csv
```

```
df=pd.read_csv('/kaggle/input/play-tennis/play_tennis.csv')
df
```

	day	outlook	temp	humidity	wind	play
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

```
#problem 1

#outlook = Sunny temp=hot humidity=high wind=weak
#play tennis or not


#solution 1

#probability=number of favourable outcomes/total number of outcomes

# p(yes | sunny,hot,high,weak)=p(sunny|yes)*p(hot|yes)*p(high|yes)*p(weak|yes)*p(yes)
```

```
#p(no | sunny,hot,high,weak)=p(sunny|no)*p(hot|no)*p(high|no)*p(weak|no)*p(no)

# if p(yes | sunny,hot,high,weak) > p(no | sunny,hot,high,weak) => then yes they will play tennis otherwise not play tennis
```

```
# training phase

# in training phase we make a lookup table(dictionary)=> where we have stored all the possible probabilities of all possibl
```

```
#testing phase
```

```
#training phase

# P(yes)

# P(no)

#to find the count of yes and no

df['play'].value_counts()
```

```
play
Yes    9
No     5
Name: count, dtype: int64
```

```
#probability of yes
prob_y=9/14
print("probab of yes: ",prob_y)

#probability of no
prob_n=5/14
print("probab of no: ",prob_n)
```

```
probab of yes:  0.6428571428571429
probab of no:  0.35714285714285715
```

```
df[(df['play']=="Yes") & (df['outlook']=="Sunny")]
```

	day	outlook	temp	humidity	wind	play
8	D9	Sunny	Cool	Normal	Weak	Yes
10	D11	Sunny	Mild	Normal	Strong	Yes

```
df[(df['play']=="Yes") & (df['humidity']=="High")]
```

	day	outlook	temp	humidity	wind	play
2	D3	Overcast	Hot	High	Weak	Yes
3	D4	Rain	Mild	High	Weak	Yes
11	D12	Overcast	Mild	High	Strong	Yes

```
df[(df['play']=="Yes") & (df['wind']=="Weak")]
```

	day	outlook	temp	humidity	wind	play
2	D3	Overcast	Hot	High	Weak	Yes
3	D4	Rain	Mild	High	Weak	Yes
4	D5	Rain	Cool	Normal	Weak	Yes
8	D9	Sunny	Cool	Normal	Weak	Yes
9	D10	Rain	Mild	Normal	Weak	Yes
12	D13	Overcast	Hot	Normal	Weak	Yes

```
df[df['play']=='Yes']
```

	day	outlook	temp	humidity	wind	play
2	D3	Overcast	Hot	High	Weak	Yes
3	D4	Rain	Mild	High	Weak	Yes
4	D5	Rain	Cool	Normal	Weak	Yes
6	D7	Overcast	Cool	Normal	Strong	Yes
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

```
df[(df['play']=="Yes") & (df['temp']=="Hot")]
```

	day	outlook	temp	humidity	wind	play
2	D3	Overcast	Hot	High	Weak	Yes
12	D13	Overcast	Hot	Normal	Weak	Yes

```
#outlook

#crosstab(row,column)=> it will analyze the relationship between the variables and will create contignecy tables.
# it help us to analyze how often the combinations of values occur

pd.crosstab(df['outlook'],df['play'])
# pd.crosstab(df['play'],df['outlook'])
```

play	No	Yes
outlook		
Overcast	0	4
Rain	2	3
Sunny	3	2

```
#probability for given that they will not play tennis when it is overcast
pon=0

#probability for given that they will not play tennis when it is raining
prn=2/5

#probability for given that they will not play tennis when it is sunny
psn=3/5

#probability for given that they will  play tennis when it is overcast
poy=4/9

#probability for given that they will play tennis when it is raining
pry=3/9

#probability for given that they will play tennis when it is sunny
psy=2/9



#temperature
pd.crosstab(df['temp'],df['play'])
```

play	No	Yes
temp		
Cool	1	3
Hot	2	2
Mild	2	4

```
#probability of cold no
pcn=1/5

#probability of hot no
phn=2/5

#probability of mild no
pmn=2/5


#probability of cold yes
pcy=3/9

#probability of hot yes
phy=2/9

#probability of mild yes
pmy=4/9
```

```
#humidity

pd.crosstab(df['humidity'],df['play'])
```

play	No	Yes
humidity		
High	4	3
Normal	1	6

```
# probability of high no
phighn=4/5


# probability of normal no
pnn=1/5

# probability of high yes
phighy=3/9

# probability of normal yes
pny=6/9


#wind

pd.crosstab(df['wind'],df['play'])
```

play	No	Yes
wind		
Strong	3	3
Weak	2	6

```
#probability of strong nc
pstrongn=3/5

#probability of weak no
```

```
pwn=2/5

#probability of strong ye
pstrongy=3/9

#probability of weak yes
pwy=6/9
```

```
#outlook = Sunny temp=hot humidity=high wind=weak
#ans=p(sunny|yes)*p(hot|yes)*p(high|yes)*p(weak|yes)*p(yes)
ansYes=psy*phy*phighy*pwy*prob_y
print("first probability:",ansYes)
```

```
first probability: 0.007054673721340387
```

```
#p(no | sunny,hot,high,weak)=p(sunny|no)*p(hot|no)*p(high|no)*p(weak|no)*p(no)
ansNo=psn*phn*phighn*pwn*prob_n
print(ansNo)
```