Naive Bayes - Categorical Data

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:
```
dataset = pd.read_csv('http://people.bu.edu/kalathur/datasets/weather.csv')
```

In [3]:
```
dataset
```

Out[3]:

	outlook	temperature	humidity	windy	play
0	sunny	hot	high	False	no
1	sunny	hot	high	True	no
2	overcast	hot	high	False	yes
3	rainy	mild	high	False	yes
4	rainy	cool	normal	False	yes
5	rainy	cool	normal	True	no
6	overcast	cool	normal	True	yes
7	sunny	mild	high	False	no
8	sunny	cool	normal	False	yes
9	rainy	mild	normal	False	yes
10	sunny	mild	normal	True	yes
11	overcast	mild	high	True	yes
12	overcast	hot	normal	False	yes
13	rainy	mild	high	True	no

In [4]:
```
dataset['play'].value_counts()
```

Out[4]:
```
yes    9
no     5
Name: play, dtype: int64
```

In [5]:
```
dataset['outlook'].unique()
```

Out[5]:
```
array(['sunny', 'overcast', 'rainy'], dtype=object)
```

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

- Encode the labels to numeric values

In [7]:
```
df = dataset.copy()
```

In [8]:
```
le = {}

for col in df.columns:
    le[col] = preprocessing.LabelEncoder()
    le[col].fit(df[col].unique())
    print('{0:12s} => {1}'.format(col, le[col].classes_))
    df[col] = le[col].transform(df[col])

df
```

```
outlook      => ['overcast' 'rainy' 'sunny']
temperature  => ['cool' 'hot' 'mild']
humidity     => ['high' 'normal']
windy        => [False  True]
play         => ['no' 'yes']
```

Out[8]:

	outlook	temperature	humidity	windy	play
0	2	1	0	0	0
1	2	1	0	1	0
2	0	1	0	0	1
3	1	2	0	0	1
4	1	0	1	0	1
5	1	0	1	1	0
6	0	0	1	1	1
7	2	2	0	0	0
8	2	0	1	0	1
9	1	2	1	0	1
10	2	2	1	1	1
11	0	2	0	1	1
12	0	1	1	0	1
13	1	2	0	1	0

In [9]:
```
le
```

Out[9]:
```
{'outlook': LabelEncoder(),
 'temperature': LabelEncoder(),
 'humidity': LabelEncoder(),
 'windy': LabelEncoder(),
 'play': LabelEncoder()}
```

In [10]:
```
from sklearn import metrics
from sklearn.naive_bayes import GaussianNB
```

In [11]:
```
gnb = GaussianNB()
```

In [12]:
```
used_features =[
    "outlook",
    "temperature",
    "humidity",
    "windy"
]

# Train classifier
gnb.fit(
    df[used_features].values,
    df["play"]
)
```

Out[12]:
```
GaussianNB(priors=None, var_smoothing=1e-09)
```

In [13]:
```
y_pred = gnb.predict(df[used_features])

# Print results
print("Number of mislabeled points out of a total {} points : {}, performance {:05.2f}%"
      .format(
          df.shape[0],
          (df["play"] != y_pred).sum(),
          100*(1-(df["play"] != y_pred).sum()/df.shape[0])
))
```

```
Number of mislabeled points out of a total 14 points : 1, performance 92.86%
```

In [14]:
```
pd.DataFrame({'predicted': le['play'].inverse_transform(y_pred), 'actual': dataset['play']})
```

Out[14]:

	predicted	actual
0	no	no
1	no	no
2	yes	yes
3	yes	yes
4	yes	yes
5	yes	no
6	yes	yes
7	no	no
8	yes	yes
9	yes	yes
10	yes	yes
11	yes	yes
12	yes	yes
13	no	no

In [15]:
```
metrics.confusion_matrix(y_pred, df['play'])
```

Out[15]:
```
array([[4, 0],
       [1, 9]])
```

In [ ]:
```

```