NAME: CHETAN SALMOTRA

ROLL NO: 142

PRN: 0120180135

In [1]:

```
## Importing neccessary lib required
import pandas as pd
from sklearn.preprocessing import LabelEncoder
```

In [2]:

```
df=pd.read_csv('weather.csv')
```

In [3]:

```
df.head()
```

Out[3]:

	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDir	WindGustSpeed	WindDir9am	WindDir3pm	WindSpeed
0	8.0	24.3	0.0	3.4	6.3	NW	30.0	SW	NW	
1	14.0	26.9	3.6	4.4	9.7	ENE	39.0	E	W	
2	13.7	23.4	3.6	5.8	3.3	NW	85.0	N	NNE	
3	13.3	15.5	39.8	7.2	9.1	NW	54.0	WNW	W	
4	7.6	16.1	2.8	5.6	10.6	SSE	50.0	SSE	ESE	

5 rows × 22 columns

In [4]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 366 entries, 0 to 365
Data columns (total 22 columns):
MinTemp          366 non-null float64
MaxTemp          366 non-null float64
Rainfall         366 non-null float64
Evaporation      366 non-null float64
Sunshine         363 non-null float64
WindGustDir      363 non-null object
WindGustSpeed    364 non-null float64
WindDir9am       335 non-null object
WindDir3pm       365 non-null object
WindSpeed9am     359 non-null float64
WindSpeed3pm     366 non-null int64
Humidity9am      366 non-null int64
Humidity3pm      366 non-null int64
Pressure9am      366 non-null float64
Pressure3pm      366 non-null float64
Cloud9am         366 non-null int64
Cloud3pm         366 non-null int64
Temp9am          366 non-null float64
Temp3pm          366 non-null float64
RainToday        366 non-null object
RISK_MM          366 non-null float64
RainTomorrow     366 non-null object
```

```
dtypes: float64(12), int64(5), object(5)
memory usage: 63.0+ KB
```

In [7]:

```
le=LabelEncoder()
df['WindGustDir']= le.fit_transform(df['WindGustDir'].astype(str))
df['WindDir9am']= le.fit_transform(df['WindDir9am'].astype(str))
df['WindDir3pm']= le.fit_transform(df['WindDir3pm'].astype(str))
df['RainToday']= le.fit_transform(df['RainToday'].astype(str))
df.head()
```

Out[7]:

	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDir	WindGustSpeed	WindDir9am	WindDir3pm	WindSpeed9
0	8.0	24.3	0.0	3.4	6.3	7	30.0	12	7	
1	14.0	26.9	3.6	4.4	9.7	1	39.0	0	13	
2	13.7	23.4	3.6	5.8	3.3	7	85.0	3	5	
3	13.3	15.5	39.8	7.2	9.1	7	54.0	14	13	
4	7.6	16.1	2.8	5.6	10.6	10	50.0	10	2	

5 rows × 22 columns

In [9]:

```
df['Sunshine'].fillna(df['Sunshine'].mean(),inplace=True)
df['WindGustSpeed'].fillna(df['WindGustSpeed'].mean(),inplace=True)
df['WindSpeed9am'].fillna(df['WindSpeed9am'].value_counts().index[0],inplace=True)
```

In [10]:

```
df.isnull().sum()
```

Out[10]:

```
MinTemp          0
MaxTemp          0
Rainfall         0
Evaporation      0
Sunshine         0
WindGustDir      0
WindGustSpeed    0
WindDir9am       0
WindDir3pm       0
WindSpeed9am     0
WindSpeed3pm     0
Humidity9am      0
Humidity3pm      0
Pressure9am      0
Pressure3pm      0
Cloud9am         0
Cloud3pm         0
Temp9am          0
Temp3pm          0
RainToday        0
RISK_MM          0
RainTomorrow     0
dtype: int64
```

In [11]:

```
X = df[['MinTemp', 'MaxTemp', 'Rainfall', 'Evaporation', 'Sunshine',
       'WindGustDir', 'WindGustSpeed', 'WindDir9am', 'WindDir3pm',
       'WindSpeed9am', 'WindSpeed3pm', 'Humidity9am', 'Humidity3pm',
       'Pressure9am', 'Pressure3pm', 'Cloud9am', 'Cloud3pm', 'Temp9am',
       'Temp3pm','RainToday', 'RISK_MM']]
```

```
y = df[['RainTomorrow']]
```

In [12]:

```
from sklearn.model_selection import train_test_split
```

In [13]:

```
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.20,random_state=0)
```

In [14]:

```
from sklearn.naive_bayes import GaussianNB
```

In [15]:

```
clf = GaussianNB()
```

In [16]:

```
clf.fit(X_train,y_train)
```

```
D:\anaconda\lib\site-packages\sklearn\utils\validation.py:761: DataConversionWarning: A c
olumn-vector y was passed when a 1d array was expected. Please change the shape of y to (
n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

Out[16]:

```
GaussianNB(priors=None, var_smoothing=1e-09)
```

In [17]:

```
y_pred = clf.predict(X_test)
```

In [18]:

```
# This is what we are getting on test data set

y_pred
```

Out[18]:

```
array(['No', 'No', 'Yes', 'No', 'Yes', 'Yes', 'Yes', 'Yes', 'No', 'No',
       'No', 'No', 'Yes', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'Yes',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'Yes', 'No',
       'Yes', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No', 'No', 'No',
       'Yes', 'No', 'No', 'Yes', 'No', 'No', 'Yes', 'Yes', 'No', 'No',
       'No', 'Yes', 'No', 'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'No'],
      dtype='<U3')
```

In [19]:

```
from sklearn.metrics import confusion_matrix,accuracy_score,classification_report
```

In [20]:

```
"""
Confusion Matrix
"""
print(confusion_matrix(y_test,y_pred))
```

```
[[55  6]
 [ 0 13]]
```

In [21]:

```
"""
Confusion Matrix : Result
"""
```

```
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

          No       1.00      0.90      0.95        61
         Yes       0.68      1.00      0.81        13

   micro avg       0.92      0.92      0.92        74
   macro avg       0.84      0.95      0.88        74
weighted avg       0.94      0.92      0.92        74
```

In [22]:

```
"""
Here we are getting Acc of 92%
"""

print("Accuracy =",accuracy_score(y_test,y_pred)*100)
```

```
Accuracy = 91.8918918918919
```