

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

```
In [2]: df1 = pd.read_csv('weather_pathankot_2017_2020 (1).csv')
```

```
In [3]: # df.head()
```

```
In [4]: df1.head()
```

Out[4]:

| | Cloud Cover | Date time | Dew Point | Relative Humidity | Precipitation Cover | Temperature | Visibility | Conditions |
|---|-------------|------------|-----------|-------------------|---------------------|-------------|------------|------------------|
| 0 | 30.0 | 01-01-2017 | 6.9 | 59.99 | 0.0 | 14.6 | 4.0 | Partially cloudy |
| 1 | 25.0 | 01-02-2017 | 6.9 | 61.28 | 50.0 | 14.2 | 4.0 | Rain |
| 2 | 90.0 | 01-03-2017 | 7.0 | 63.79 | 0.0 | 13.8 | 4.0 | Overcast |
| 3 | 100.0 | 01-04-2017 | 6.1 | 62.09 | 0.0 | 13.2 | 4.0 | Overcast |
| 4 | 100.0 | 01-05-2017 | 7.7 | 69.75 | 0.0 | 13.1 | 4.0 | Overcast |

```
In [5]: df1.dropna(inplace=True)
```

```
In [6]: df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2710 entries, 0 to 2737
Data columns (total 8 columns):
Cloud Cover          2710 non-null float64
Date time            2710 non-null object
Dew Point            2710 non-null float64
Relative Humidity     2710 non-null float64
Precipitation Cover  2710 non-null float64
Temperature          2710 non-null float64
Visibility           2710 non-null float64
Conditions            2710 non-null object
dtypes: float64(6), object(2)
memory usage: 190.5+ KB
```

```
In [7]: df1['Precipitation Cover'] = df1['Precipitation Cover'].round(decimals=0)
```

```
In [8]: df1["Precipitation Cover"] = df1["Precipitation Cover"].astype(int)
```

```
In [9]: df1['Precipitation Cover'] = df1['Precipitation Cover'].replace([0,4,5,6,7,8,9,10], 0)
df1['Precipitation Cover'] = df1['Precipitation Cover'].replace([11,12,13,14,15,16,17,18,19,21,26,33,50], 1)
df1['Precipitation Cover'] = df1['Precipitation Cover'].replace([67,100], 2)
```

```
In [10]: df1['Precipitation Cover'].value_counts()
```

```
Out[10]: 0    2137
         1     343
         2     230
         Name: Precipitation Cover, dtype: int64
```

```
In [11]: df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2710 entries, 0 to 2737
Data columns (total 8 columns):
Cloud Cover          2710 non-null float64
Date time            2710 non-null object
Dew Point            2710 non-null float64
Relative Humidity     2710 non-null float64
Precipitation Cover  2710 non-null int32
Temperature          2710 non-null float64
Visibility           2710 non-null float64
Conditions           2710 non-null object
dtypes: float64(5), int32(1), object(2)
memory usage: 180.0+ KB
```

```
In [12]: df1.head()
```

```
Out[12]:
```

| | Cloud Cover | Date time | Dew Point | Relative Humidity | Precipitation Cover | Temperature | Visibility | Conditions |
|---|-------------|------------|-----------|-------------------|---------------------|-------------|------------|------------------|
| 0 | 30.0 | 01-01-2017 | 6.9 | 59.99 | 0 | 14.6 | 4.0 | Partially cloudy |
| 1 | 25.0 | 01-02-2017 | 6.9 | 61.28 | 1 | 14.2 | 4.0 | Rain |
| 2 | 90.0 | 01-03-2017 | 7.0 | 63.79 | 0 | 13.8 | 4.0 | Overcast |
| 3 | 100.0 | 01-04-2017 | 6.1 | 62.09 | 0 | 13.2 | 4.0 | Overcast |
| 4 | 100.0 | 01-05-2017 | 7.7 | 69.75 | 0 | 13.1 | 4.0 | Overcast |

```
In [13]: df1["Cloud Cover"] = df1["Cloud Cover"].astype(int)
```

```
In [14]: df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2710 entries, 0 to 2737
Data columns (total 8 columns):
Cloud Cover          2710 non-null int32
Date time            2710 non-null object
Dew Point            2710 non-null float64
Relative Humidity     2710 non-null float64
Precipitation Cover  2710 non-null int32
Temperature          2710 non-null float64
Visibility           2710 non-null float64
Conditions           2710 non-null object
dtypes: float64(4), int32(2), object(2)
memory usage: 169.4+ KB
```

```
In [15]: #dict(df1["Cloud Cover"].value_counts())
```

```
In [16]: df1['Cloud Cover'] = df1['Cloud Cover'].replace([0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25], 0)
df1['Cloud Cover'] = df1['Cloud Cover'].replace([26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50], 1)
df1['Cloud Cover'] = df1['Cloud Cover'].replace([51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75], 2)
df1['Cloud Cover'] = df1['Cloud Cover'].replace([76,77,80,82,84,85,86,87,88,90,91,93,95,96,98,100], 3)
```

```
In [17]: df1.Visibility.unique()
```

```
Out[17]: array([ 4. ,  2. ,  3. ,  3.3,  8.5, 11.5, 12. ,  2.5,  1. ,  1.5,  0.8,
        0.3,  1.6,  2.3,  3.1,  2.4,  1.9,  3.7,  3.8,  4.2,  3.2,  1.4,
        2.7,  2.9,  3.9,  3.5,  2.6,  2.8,  4.1,  1.7,  4.3,  4.4,  4.5,
        5. ,  3.6,  4.7,  3.4,  4.8,  4.6,  5.1,  4.9,  2.1,  1.2,  1.8,
        1.1,  2.2,  0.9,  1.3,  0.7,  0.6,  0.5,  0.4,  0.2,  0.1,  5.3,
        5.2])
```

```
In [18]: df1["Visibility"] = df1["Visibility"].round(decimals=0)
```

```
In [19]: df1["Visibility"] = df1["Visibility"].astype(int)
```

```
In [20]: df1.Visibility.value_counts()
```

```
Out[20]: 4      2000
3       431
2       152
5        69
1        45
0         9
12         3
8          1
Name: Visibility, dtype: int64
```

```
In [21]: df1['Conditions'].value_counts()
```

```
Out[21]: Clear      952
Partially cloudy  898
Rain, Partially cloudy  362
Rain, Overcast    316
Overcast         148
Rain              34
Name: Conditions, dtype: int64
```

```
In [22]: df1['Conditions'] = df1['Conditions'].map({'Clear':0,
        'Partially cloudy':1,
        'Rain, Overcast':2,
        'Rain, Partially cloudy':3,
        'Overcast':4,
        'Rain':5})
```

```
In [23]: df1['Conditions'].value_counts()
```

```
Out[23]: 0    952
         1    898
         3    362
         2    316
         4    148
         5     34
         Name: Conditions, dtype: int64
```

```
In [24]: df1.head()
```

```
Out[24]:
```

| | Cloud Cover | Date time | Dew Point | Relative Humidity | Precipitation Cover | Temperature | Visibility | Conditions |
|---|-------------|------------|-----------|-------------------|---------------------|-------------|------------|------------|
| 0 | 1 | 01-01-2017 | 6.9 | 59.99 | 0 | 14.6 | 4 | 1 |
| 1 | 0 | 01-02-2017 | 6.9 | 61.28 | 1 | 14.2 | 4 | 5 |
| 2 | 3 | 01-03-2017 | 7.0 | 63.79 | 0 | 13.8 | 4 | 4 |
| 3 | 3 | 01-04-2017 | 6.1 | 62.09 | 0 | 13.2 | 4 | 4 |
| 4 | 3 | 01-05-2017 | 7.7 | 69.75 | 0 | 13.1 | 4 | 4 |

```
In [25]: df1.drop('Date time', axis=1, inplace=True)
```

```
In [26]: df1.drop('Visibility', axis=1, inplace=True)
```

```
In [27]: df1.drop('Dew Point', axis=1, inplace=True)
```

```
In [28]: df1.head()
```

```
Out[28]:
```

| | Cloud Cover | Relative Humidity | Precipitation Cover | Temperature | Conditions |
|---|-------------|-------------------|---------------------|-------------|------------|
| 0 | 1 | 59.99 | 0 | 14.6 | 1 |
| 1 | 0 | 61.28 | 1 | 14.2 | 5 |
| 2 | 3 | 63.79 | 0 | 13.8 | 4 |
| 3 | 3 | 62.09 | 0 | 13.2 | 4 |
| 4 | 3 | 69.75 | 0 | 13.1 | 4 |

```
In [29]: from sklearn.model_selection import train_test_split
X = df1.drop('Conditions',axis=1).values
Y = df1['Conditions'].values
```

```
In [30]: from sklearn.preprocessing import MinMaxScaler
minmax = MinMaxScaler()
X = minmax.fit_transform(X)
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.20, random_state=42)
print('X_train :',X_train.shape)
print('X_test :',X_test.shape)
print('Y_train :',Y_train.shape)
print('Y_test :',Y_test.shape)
```

```
X_train : (2168, 4)
X_test : (542, 4)
Y_train : (2168,)
Y_test : (542,)
```

```
In [31]: from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier(n_estimators=200)
rfc.fit(X_train,Y_train)
pr = rfc.predict(X_test)
```

```
In [32]: from sklearn.metrics import r2_score,confusion_matrix
r = r2_score(Y_test,pr)
print("R2score when we predict using Randomn forest is ",r)
```

```
R2score when we predict using Randomn forest is 0.8636007754828127
```

```
In [33]: Rain_Conditions = {0:'Clear',1:'Partially cloudy',2:'Rain, Overcast',3:'Rain, Partially cloudy',4:'Overcast',5:'Rain'}
```

```
In [34]: df1.head()
```

Out[34]:

| | Cloud Cover | Relative Humidity | Precipitation Cover | Temperature | Conditions |
|---|-------------|-------------------|---------------------|-------------|------------|
| 0 | 1 | 59.99 | 0 | 14.6 | 1 |
| 1 | 0 | 61.28 | 1 | 14.2 | 5 |
| 2 | 3 | 63.79 | 0 | 13.8 | 4 |
| 3 | 3 | 62.09 | 0 | 13.2 | 4 |
| 4 | 3 | 69.75 | 0 | 13.1 | 4 |

```
In [35]: pred_val = rfc.predict([[0,69.80,0,39]])
```

```
In [36]: pred_val = round(int(pred_val))
```

```
In [37]: print(Rain_Conditions[pred_val])
```

```
Clear
```

```
In [38]: Rain_Conditions.keys()
```

```
Out[38]: dict_keys([0, 1, 2, 3, 4, 5])
```

```
In [39]: prd_val = rfc.predict([[0,25,0,33]])
```

```
In [40]: ped_val = round(int(prd_val))
```

```
In [41]: print(Rain_Conditions[ped_val])
```

Clear

```
In [42]: df1['Precipitation Cover'].unique()  
df1['Cloud Cover'].unique()
```

```
Out[42]: array([1, 0, 3, 2], dtype=int64)
```

```
In [43]: ex = np.array([0,70.3,0,12]).reshape(1,-1)  
val = round(int(rfc.predict(ex)))  
print(Rain_Conditions[val])
```

Clear