```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df=pd.read_csv("weather.csv")

print(df)
```

```
     MinTemp  MaxTemp  Rainfall  Evaporation  Sunshine WindGustDir  \
0        8.0     24.3       0.0          3.4       6.3          NW
1       14.0     26.9       3.6          4.4       9.7         ENE
2       13.7     23.4       3.6          5.8       3.3          NW
3       13.3     15.5      39.8          7.2       9.1          NW
4        7.6     16.1       2.8          5.6      10.6         SSE
..       ...      ...       ...          ...       ...         ...
361      9.0     30.7       0.0          7.6      12.1         NNW
362      7.1     28.4       0.0         11.6      12.7           N
363     12.5     19.9       0.0          8.4       5.3         ESE
364     12.5     26.9       0.0          5.0       7.1          NW
365     12.3     30.2       0.0          6.0      12.6          NW

     WindGustSpeed WindDir9am WindDir3pm  WindSpeed9am  ...
Humidity3pm  \
0             30.0         SW         NW           6.0  ...
29
1             39.0          E          W           4.0  ...
36
2             85.0          N        NNE           6.0  ...
69
3             54.0        WNW          W          30.0  ...
56
4             50.0        SSE        ESE          20.0  ...
49
..             ...        ...        ...           ...  ...           .
..
361           76.0        SSE         NW           7.0  ...
15
362           48.0        NNW        NNW           2.0  ...
22
363           43.0        ENE        ENE          11.0  ...
47
364           46.0        SSW        WNW           6.0  ...
39
365           78.0         NW        WNW          31.0  ...
13

     Pressure9am  Pressure3pm  Cloud9am  Cloud3pm  Temp9am  Temp3pm  \
0         1019.7       1015.0         7         7     14.4     23.6
1         1012.4       1008.4         5         3     17.5     25.7
2         1009.5       1007.2         8         7     15.4     20.2
```

```
3          1005.5       1007.0          2          7     13.5     14.1
4          1018.3       1018.5          7          7     11.1     15.4
..            ...          ...        ...        ...      ...      ...
361        1016.1       1010.8          1          3     20.4     30.0
362        1020.0       1016.9          0          1     17.2     28.2
363        1024.0       1022.8          3          2     14.5     18.3
364        1021.0       1016.2          6          7     15.8     25.9
365        1009.6       1009.2          1          1     23.8     28.6

     RainToday  RISK_MM RainTomorrow
0           No      3.6          Yes
1          Yes      3.6          Yes
2          Yes     39.8          Yes
3          Yes      2.8          Yes
4          Yes      0.0           No
..         ...      ...          ...
361         No      0.0           No
362         No      0.0           No
363         No      0.0           No
364         No      0.0           No
365         No      0.0           No

[366 rows x 22 columns]
```

```
print(df.columns)
```

```
Index(['MinTemp', 'MaxTemp', 'Rainfall', 'Evaporation', 'Sunshine',
       'WindGustDir', 'WindGustSpeed', 'WindDir9am', 'WindDir3pm',
       'WindSpeed9am', 'WindSpeed3pm', 'Humidity9am', 'Humidity3pm',
       'Pressure9am', 'Pressure3pm', 'Cloud9am', 'Cloud3pm',
'Temp9am',
       'Temp3pm', 'RainToday', 'RISK_MM', 'RainTomorrow'],
      dtype='object')
```

```
print(df.head())
print(df.shape)
print(df.info())
print(df.describe())
print(df.isnull().sum())
```

```
   MinTemp  MaxTemp  Rainfall  Evaporation  Sunshine WindGustDir  \
0      8.0     24.3       0.0          3.4       6.3          NW
1     14.0     26.9       3.6          4.4       9.7         ENE
2     13.7     23.4       3.6          5.8       3.3          NW
3     13.3     15.5      39.8          7.2       9.1          NW
4      7.6     16.1       2.8          5.6      10.6         SSE

   WindGustSpeed WindDir9am WindDir3pm  WindSpeed9am  ...  Humidity3pm
\
0           30.0         SW         NW           6.0  ...           29
```

```
1           39.0          E           W           4.0  ...           36

2           85.0          N         NNE           6.0  ...           69

3           54.0        WNW           W          30.0  ...           56

4           50.0        SSE         ESE          20.0  ...           49


   Pressure9am  Pressure3pm  Cloud9am  Cloud3pm  Temp9am  Temp3pm
RainToday  \
0       1019.7       1015.0         7         7     14.4     23.6
No
1       1012.4       1008.4         5         3     17.5     25.7
Yes
2       1009.5       1007.2         8         7     15.4     20.2
Yes
3       1005.5       1007.0         2         7     13.5     14.1
Yes
4       1018.3       1018.5         7         7     11.1     15.4
Yes

   RISK_MM RainTomorrow
0      3.6          Yes
1      3.6          Yes
2     39.8          Yes
3      2.8          Yes
4      0.0           No

[5 rows x 22 columns]
(366, 22)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 366 entries, 0 to 365
Data columns (total 22 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   MinTemp        366 non-null    float64
 1   MaxTemp        366 non-null    float64
 2   Rainfall       366 non-null    float64
 3   Evaporation    366 non-null    float64
 4   Sunshine       363 non-null    float64
 5   WindGustDir    363 non-null    object
 6   WindGustSpeed  364 non-null    float64
 7   WindDir9am     335 non-null    object
 8   WindDir3pm     365 non-null    object
 9   WindSpeed9am   359 non-null    float64
 10  WindSpeed3pm   366 non-null    int64
 11  Humidity9am    366 non-null    int64
 12  Humidity3pm    366 non-null    int64
 13  Pressure9am    366 non-null    float64
```

```
 14  Pressure3pm    366 non-null    float64
 15  Cloud9am       366 non-null    int64
 16  Cloud3pm       366 non-null    int64
 17  Temp9am        366 non-null    float64
 18  Temp3pm        366 non-null    float64
 19  RainToday      366 non-null    object
 20  RISK_MM        366 non-null    float64
 21  RainTomorrow   366 non-null    object
dtypes: float64(12), int64(5), object(5)
memory usage: 63.0+ KB
None
          MinTemp     MaxTemp    Rainfall  Evaporation    Sunshine  \
count  366.000000  366.000000  366.000000   366.000000  363.000000
mean     7.265574   20.550273    1.428415     4.521858    7.909366
std      6.025800    6.690516    4.225800     2.669383    3.481517
min     -5.300000    7.600000    0.000000     0.200000    0.000000
25%      2.300000   15.025000    0.000000     2.200000    5.950000
50%      7.450000   19.650000    0.000000     4.200000    8.600000
75%     12.500000   25.500000    0.200000     6.400000   10.500000
max     20.900000   35.800000   39.800000    13.800000   13.600000

       WindGustSpeed  WindSpeed9am  WindSpeed3pm  Humidity9am
Humidity3pm  \
count     364.000000    359.000000    366.000000   366.000000
366.000000
mean       39.840659      9.651811     17.986339    72.035519
44.519126
std        13.059807      7.951929      8.856997    13.137058
16.850947
min        13.000000      0.000000      0.000000    36.000000
13.000000
25%        31.000000      6.000000     11.000000    64.000000
32.250000
50%        39.000000      7.000000     17.000000    72.000000
43.000000
75%        46.000000     13.000000     24.000000    81.000000
55.000000
max        98.000000     41.000000     52.000000    99.000000
96.000000

       Pressure9am  Pressure3pm    Cloud9am    Cloud3pm     Temp9am  \
count   366.000000   366.000000  366.000000  366.000000  366.000000
mean   1019.709016  1016.810383    3.890710    4.024590   12.358470
std       6.686212     6.469422    2.956131    2.666268    5.630832
min     996.500000   996.800000    0.000000    0.000000    0.100000
25%    1015.350000  1012.800000    1.000000    1.000000    7.625000
50%    1020.150000  1017.400000    3.500000    4.000000   12.550000
75%    1024.475000  1021.475000    7.000000    7.000000   17.000000
max    1035.700000  1033.200000    8.000000    8.000000   24.700000
```

```
          Temp3pm     RISK_MM
count  366.000000  366.000000
mean    19.230874    1.428415
std      6.640346    4.225800
min      5.100000    0.000000
25%     14.150000    0.000000
50%     18.550000    0.000000
75%     24.000000    0.200000
max     34.500000   39.800000
MinTemp           0
MaxTemp           0
Rainfall          0
Evaporation       0
Sunshine          3
WindGustDir       3
WindGustSpeed     2
WindDir9am       31
WindDir3pm        1
WindSpeed9am      7
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
```

```
df=df.dropna()
```

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

```
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
```

```
df['RainTomorrow'].unique()
```

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

```
Y=df.RainTomorrow

print(Y.head())
```

```
0    Yes
1    Yes
2    Yes
3    Yes
4     No
Name: RainTomorrow, dtype: object
```

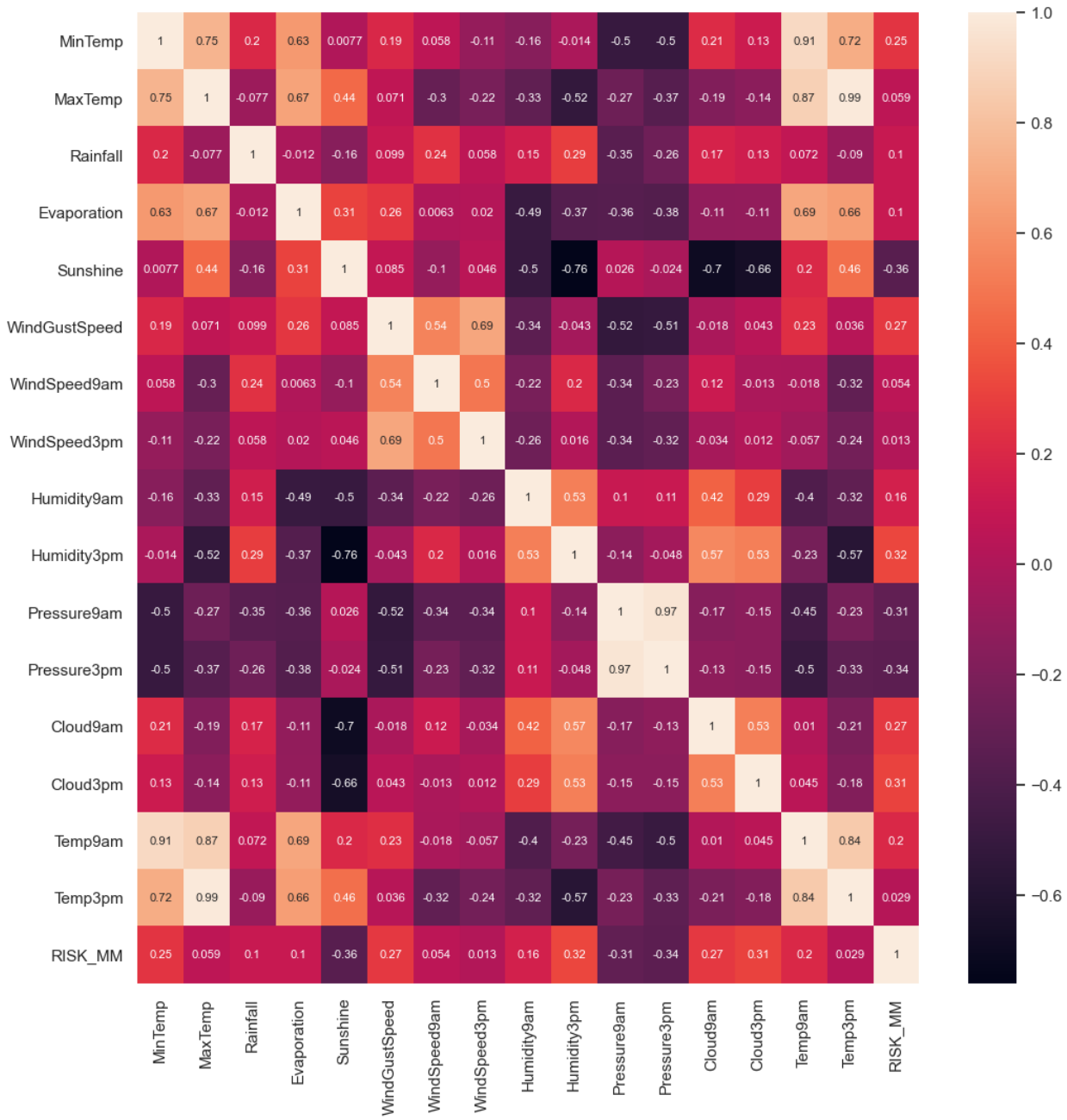
```
from sklearn import preprocessing
label_encoder=preprocessing.LabelEncoder()
df['RainTomorrow']=label_encoder.fit_transform(df['RainTomorrow'])
print(df['RainTomorrow'].unique())
label_encoder=preprocessing.LabelEncoder()
df['WindGustDir']=label_encoder.fit_transform(df['WindGustDir'])
print(df['WindGustDir'].unique())
label_encoder=preprocessing.LabelEncoder()
df['RainToday']=label_encoder.fit_transform(df['RainToday'])
print(df['RainToday'].unique())
```

```
[1 0]
[ 7  1 10  9  0  8  3 14  2  4  5  6 12 13 15 11]
[0 1]
```

```
numeric_columns = df.select_dtypes(include=['float64',
'int64']).columns
correlation_matrix = df[numeric_columns].corr()

hm = sns.heatmap(data=correlation_matrix, annot=True,
annot_kws={'size': 8})
sns.set(rc={'figure.figsize': (12, 12)})
plt.show()
```

```
# Drop non-numeric columns before further processing
X = df.drop(columns=['WindGustDir', 'RainToday', 'RainTomorrow'])
```




```
print(X.head())
```

```
   MinTemp  MaxTemp  Rainfall  Evaporation  Sunshine  WindGustSpeed  \
0      8.0     24.3       0.0          3.4       6.3           30.0
1     14.0     26.9       3.6          4.4       9.7           39.0
2     13.7     23.4       3.6          5.8       3.3           85.0
3     13.3     15.5      39.8          7.2       9.1           54.0
```

```
4      7.6     16.1       2.8           5.6      10.6            50.0

  WindDir9am WindDir3pm  WindSpeed9am  WindSpeed3pm  Humidity9am
Humidity3pm  \
0         SW         NW           6.0            20           68
29
1          E          W           4.0            17           80
36
2          N        NNE           6.0             6           82
69
3        WNW          W          30.0            24           62
56
4        SSE        ESE          20.0            28           68
49

   Pressure9am  Pressure3pm  Cloud9am  Cloud3pm  Temp9am  Temp3pm
RISK_MM
0       1019.7       1015.0         7         7     14.4     23.6
3.6
1       1012.4       1008.4         5         3     17.5     25.7
3.6
2       1009.5       1007.2         8         7     15.4     20.2
39.8
3       1005.5       1007.0         2         7     13.5     14.1
2.8
4       1018.3       1018.5         7         7     11.1     15.4
0.0
```

```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder
from sklearn import svm

# Assuming df is your DataFrame and Y is your target variable

# Select features and target variable
X = df.drop(columns=['WindGustDir', 'RainToday', 'RainTomorrow'])
Y = df['RainTomorrow']

# Perform one-hot encoding
X_encoded = pd.get_dummies(X)

# Split the data into training and test sets
X_train, X_test, Y_train, Y_test = train_test_split(X_encoded, Y,
test_size=0.2, random_state=10)

# Train the SVM model
clf = svm.SVC(kernel='linear')
clf.fit(X_train, Y_train)
```

```
# Make predictions
Y_pred = clf.predict(X_test)

from sklearn import metrics
print("Accuracy:",metrics.accuracy_score(Y_test,Y_pred))
model=svm.SVC(kernel='poly')
model.fit(X_train,Y_train)
Y_pred=model.predict(X_test)
```

```
Accuracy: 0.9696969696969697
```

```
from sklearn import metrics
print("Accuracy:",metrics.accuracy_score(Y_test,Y_pred))
```

```
Accuracy: 0.9090909090909091
```