In [24]:

```
import pandas as pd
import numpy as np
```

In [25]:

```
import seaborn as sns
import matplotlib.pyplot as plt
```

In [26]:

```
from sklearn import preprocessing
from sklearn import model_selection
from sklearn import metrics
from sklearn import linear_model
from sklearn import ensemble
from sklearn import tree
from sklearn import svm
import xgboost
```

In [27]:

```
data = pd.read_csv("E:\IBM_Project\weatherAUS.csv")
```

In [28]:

```
data.head()
```

Out[28]:

| | Date | Location | MinTemp | MaxTemp | Rainfall | Evaporation | Sunshine | WindGustDir | WindGustSpeed | WindDir9am | ... | Humidity3pm | Pressur |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 2008-12-01 | Albury | 13.4 | 22.9 | 0.6 | NaN | NaN | W | 44.0 | W | ... | 22.0 | 1 |
| **1** | 2008-12-02 | Albury | 7.4 | 25.1 | 0.0 | NaN | NaN | WNW | 44.0 | NNW | ... | 25.0 | 1 |
| **2** | 2008-12-03 | Albury | 12.9 | 25.7 | 0.0 | NaN | NaN | WSW | 46.0 | W | ... | 30.0 | 1 |
| **3** | 2008-12-04 | Albury | 9.2 | 28.0 | 0.0 | NaN | NaN | NE | 24.0 | SE | ... | 16.0 | 1 |
| **4** | 2008-12-05 | Albury | 17.5 | 32.3 | 1.0 | NaN | NaN | W | 41.0 | ENE | ... | 33.0 | 1 |

5 rows × 24 columns

In [29]:

```
data.describe()
```

Out[29]:

| | MinTemp | MaxTemp | Rainfall | Evaporation | Sunshine | WindGustSpeed | WindSpeed9am | WindSpeed3pm | Humidity9a |
|---|---|---|---|---|---|---|---|---|---|
| **count** | 141556.000000 | 141871.000000 | 140787.000000 | 81350.000000 | 74377.000000 | 132923.000000 | 140845.000000 | 139563.000000 | 140419.0000 |
| **mean** | 12.186400 | 23.226784 | 2.349974 | 5.469824 | 7.624853 | 39.984292 | 14.001988 | 18.637576 | 68.8438 |
| **std** | 6.403283 | 7.117618 | 8.465173 | 4.188537 | 3.781525 | 13.588801 | 8.893337 | 8.803345 | 19.0512 |
| **min** | -8.500000 | -4.800000 | 0.000000 | 0.000000 | 0.000000 | 6.000000 | 0.000000 | 0.000000 | 0.0000 |
| **25%** | 7.600000 | 17.900000 | 0.000000 | 2.600000 | 4.900000 | 31.000000 | 7.000000 | 13.000000 | 57.0000 |
| **50%** | 12.000000 | 22.600000 | 0.000000 | 4.800000 | 8.500000 | 39.000000 | 13.000000 | 19.000000 | 70.0000 |
| **75%** | 16.800000 | 28.200000 | 0.800000 | 7.400000 | 10.600000 | 48.000000 | 19.000000 | 24.000000 | 83.0000 |
| **max** | 33.900000 | 48.100000 | 371.000000 | 145.000000 | 14.500000 | 135.000000 | 130.000000 | 87.000000 | 100.0000 |

In [30]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 142193 entries, 0 to 142192
Data columns (total 24 columns):
 #   Column         Non-Null Count   Dtype  
---  ------         --------------   -----  
 0   Date           142193 non-null  object 
 1   Location       142193 non-null  object 
 2   MinTemp        141556 non-null  float64
 3   MaxTemp        141871 non-null  float64
 4   Rainfall       140787 non-null  float64
 5   Evaporation    81350 non-null   float64
 6   Sunshine       74377 non-null   float64
 7   WindGustDir    132863 non-null  object 
 8   WindGustSpeed  132923 non-null  float64
 9   WindDir9am     132180 non-null  object 
 10  WindDir3pm     138415 non-null  object 
 11  WindSpeed9am   140845 non-null  float64
 12  WindSpeed3pm   139563 non-null  float64
 13  Humidity9am    140419 non-null  float64
 14  Humidity3pm    138583 non-null  float64
 15  Pressure9am    128179 non-null  float64
 16  Pressure3pm    128212 non-null  float64
 17  Cloud9am       88536 non-null   float64
 18  Cloud3pm       85099 non-null   float64
 19  Temp9am        141289 non-null  float64
 20  Temp3pm        139467 non-null  float64
 21  RainToday      140787 non-null  object
 22  RISK_MM        142193 non-null  float64
 23  RainTomorrow   142193 non-null  object
dtypes: float64(17), object(7)
memory usage: 26.0+ MB
```

In [31]:

```
data.shape #gives the dimension of the data
```

Out[31]:

```
(142193, 24)
```

In [32]:

```
data.isnull().sum()
```

Out[32]:

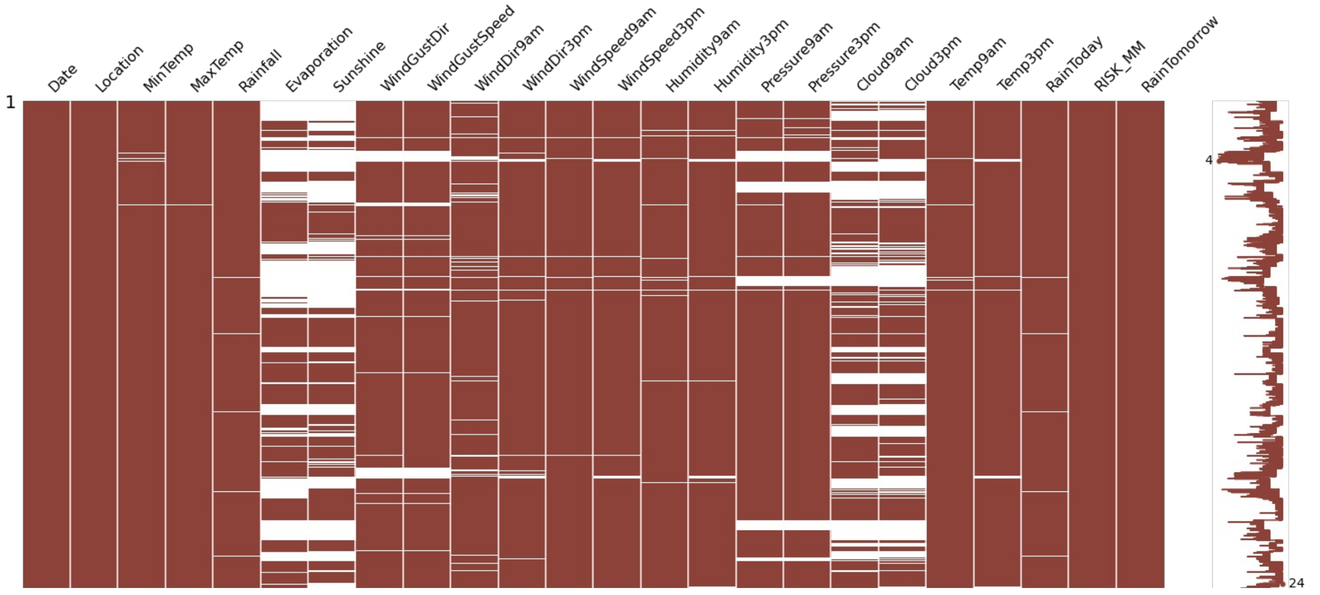
```
Date                 0
Location             0
MinTemp            637
MaxTemp            322
Rainfall          1406
Evaporation      60843
Sunshine         67816
WindGustDir       9330
WindGustSpeed     9270
WindDir9am       10013
WindDir3pm        3778
WindSpeed9am      1348
WindSpeed3pm      2630
Humidity9am       1774
Humidity3pm       3610
Pressure9am      14014
Pressure3pm      13981
Cloud9am         53657
Cloud3pm         57094
Temp9am            904
Temp3pm           2726
RainToday         1406
RISK_MM              0
RainTomorrow         0
dtype: int64
```

In [33]:

```
import missingno as msno
msno.matrix(data,color=(0.55,0.255,0.225),fontsize=16)
```

Out[33]:

```
<AxesSubplot:>
```



142193

In [34]:

```
import pandas as pd
import numpy as np
```

In [35]:

```
import seaborn as sns
import matplotlib.pyplot as plt
```

In [36]:

```
from sklearn import preprocessing
from sklearn import model_selection
from sklearn import metrics
from sklearn import linear_model
from sklearn import ensemble
from sklearn import tree
from sklearn import svm
import xgboost
```

In [37]:

```
data = pd.read_csv("E:\IBM_Project\weatherAUS.csv")
```

In [39]:

```
data_cat = data [['RainToday','WindGustDir','WindDir9am','WindDir3pm']]
data.drop(columns=['Evaporation','Sunshine','Cloud9am','Cloud3pm'],axis=1,inplace=True)
data.drop(columns=['RainToday','WindGustDir','WindDir9am','WindDir3pm'],axis=1,inplace=True)
```

In [40]:

```
# filling the missing data of numeric variables with mean
data['MinTemp'].fillna(data['MinTemp'].mean(),inplace=True)
data['MaxTemp'].fillna(data['MaxTemp'].mean(),inplace=True)
data['Rainfall'].fillna(data['Rainfall'].mean(),inplace=True)
data['WindGustSpeed'].fillna(data['WindGustSpeed'].mean(),inplace=True)
data['WindSpeed9am'].fillna(data['WindSpeed9am'].mean(),inplace=True)
data['WindSpeed3pm'].fillna(data['WindSpeed3pm'].mean(),inplace=True)
data['Humidity9am'].fillna(data['Humidity9am'].mean(),inplace=True)
data['Humidity3pm'].fillna(data['Humidity3pm'].mean(),inplace=True)
data['Pressure9am'].fillna(data['Pressure9am'].mean(),inplace=True)
data['Pressure3pm'].fillna(data['Pressure3pm'].mean(),inplace=True)
data['Temp9am'].fillna(data['Temp9am'].mean(),inplace=True)
data['Temp3pm'].fillna(data['Temp3pm'].mean(),inplace=True)
```

In [41]:

```
#filling the missing data of numeric variables with mean
cat_names=data_cat.columns
```

In [42]:

```
import numpy as np
from sklearn.impute import SimpleImputer
imp_mode=SimpleImputer(missing_values=np.nan, strategy='most_frequent')
```

In [43]:

```
data_cat=imp_mode.fit_transform(data_cat)
```

In [44]:

```
data_cat=pd.DataFrame(data_cat,columns=cat_names)
```

In [45]:

```
data=pd.concat([data,data_cat],axis=1)
```

In [ ]:

In [ ]:

In [ ]: