ex14

January 6, 2024

[50]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

import warnings
warnings.filterwarnings("ignore")
# sns.set_style("darkgrid", {"grid.color": ".6",
#                            "grid.linestyle": ":"})

from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import PolynomialFeatures
from sklearn.pipeline import make_pipeline
from sklearn.linear_model import Lasso

from xgboost import XGBClassifier
from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import GridSearchCV
```

[51]:
```
df = pd.read_csv('/Users/thutranghoa/Code/Data_analysis/Data/weatherAUS.csv.
↪download/weatherAUS.csv')
df
```

[51]:

	Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	\
0	2008-12-01	Albury	13.4	22.9	0.6	NaN	NaN	
1	2008-12-02	Albury	7.4	25.1	0.0	NaN	NaN	
2	2008-12-03	Albury	12.9	25.7	0.0	NaN	NaN	
3	2008-12-04	Albury	9.2	28.0	0.0	NaN	NaN	
4	2008-12-05	Albury	17.5	32.3	1.0	NaN	NaN	
…	…	…	…	…	…	…	…	
24553	2011-10-23	Penrith	15.1	30.7	0.0	NaN	NaN	
24554	2011-10-24	Penrith	14.9	34.5	0.0	NaN	NaN	
24555	2011-10-25	Penrith	18.9	19.9	0.4	NaN	NaN	
24556	2011-10-26	Penrith	12.3	15.9	8.2	NaN	NaN	
24557	2011-10-27	Penrith	12.3	17.0	NaN	NaN	NaN	

```
      WindGustDir  WindGustSpeed WindDir9am  … Humidity3pm  Pressure9am  \
0               W           44.0          W  …        22.0       1007.7
1             WNW           44.0        NNW  …        25.0       1010.6
2             WSW           46.0          W  …        30.0       1007.6
3              NE           24.0         SE  …        16.0       1017.6
4               W           41.0        ENE  …        33.0       1010.8
…             …             …        …  …          …          …
24553         NNE           20.0          N  …        31.0          NaN
24554          NW           39.0        WNW  …        25.0          NaN
24555           S           43.0         SE  …        98.0          NaN
24556          SE           31.0         SW  …        74.0          NaN
24557         NaN            NaN        NaN  …         NaN          NaN

       Pressure3pm  Cloud9am  Cloud3pm  Temp9am  Temp3pm  RainToday  RISK_MM  \
0           1007.1       8.0       NaN     16.9     21.8         No      0.0
1           1007.8       NaN       NaN     17.2     24.3         No      0.0
2           1008.7       NaN       2.0     21.0     23.2         No      0.0
3           1012.8       NaN       NaN     18.1     26.5         No      1.0
4           1006.0       7.0       8.0     17.8     29.7         No      0.2
…            …       …       …       …       …      …
24553          NaN       NaN       NaN     20.6     30.1         No      0.0
24554          NaN       NaN       NaN     21.1     33.8         No      0.4
24555          NaN       NaN       NaN     19.0     16.1         No      8.2
24556          NaN       NaN       NaN     13.8     15.4        Yes      1.6
24557          NaN       NaN       NaN      NaN      NaN        NaN      NaN

      RainTomorrow
0               No
1               No
2               No
3               No
4               No
…             …
24553           No
24554           No
24555          Yes
24556          Yes
24557          NaN

[24558 rows x 24 columns]
```

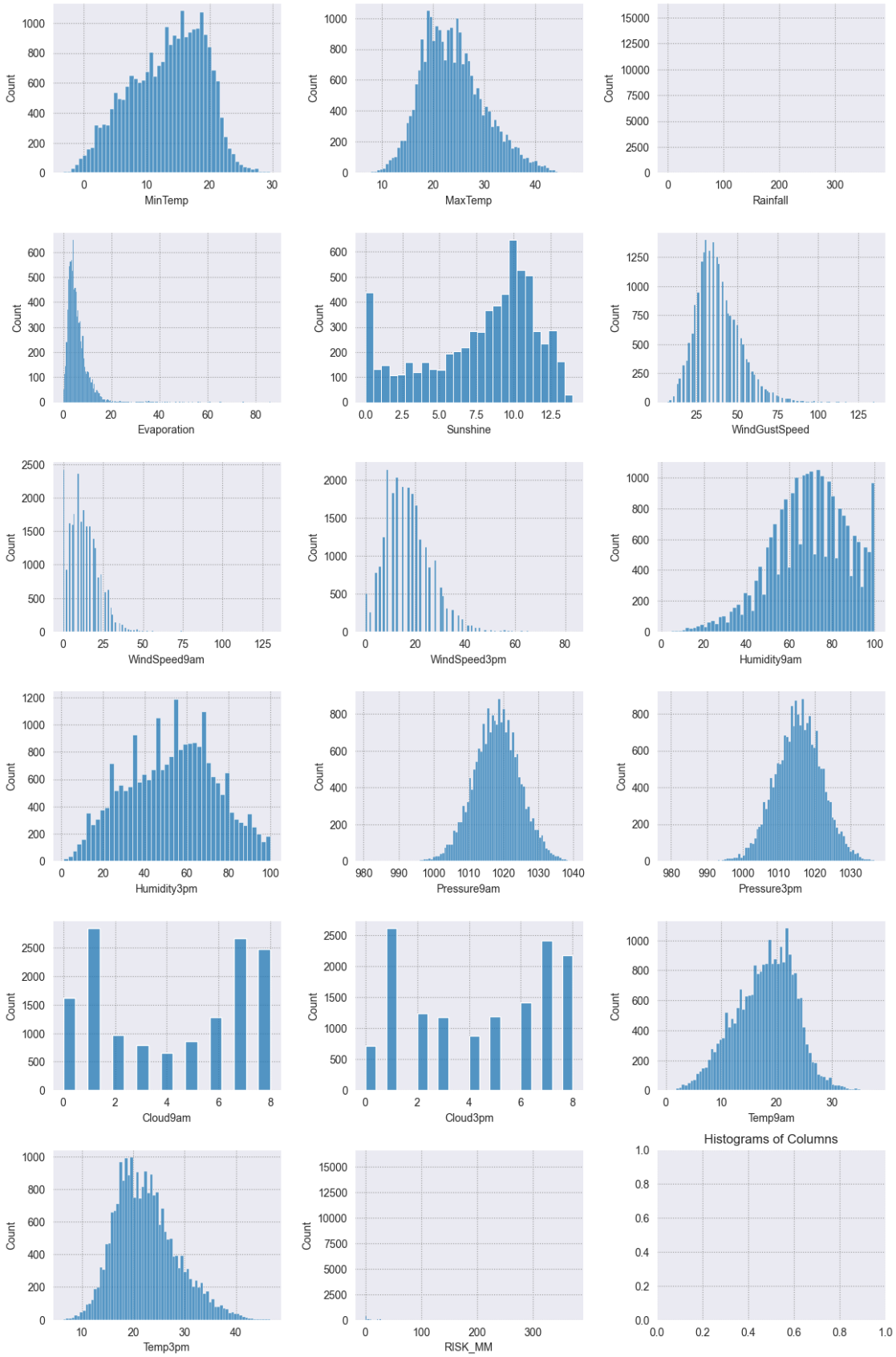
[52]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 24558 entries, 0 to 24557
Data columns (total 24 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
```

```
 0   Date           24558 non-null  object
 1   Location       24558 non-null  object
 2   MinTemp        24227 non-null  float64
 3   MaxTemp        24382 non-null  float64
 4   Rainfall       24280 non-null  float64
 5   Evaporation    9432 non-null   float64
 6   Sunshine       6664 non-null   float64
 7   WindGustDir    21100 non-null  object
 8   WindGustSpeed  21102 non-null  float64
 9   WindDir9am     21561 non-null  object
 10  WindDir3pm     22829 non-null  object
 11  WindSpeed9am   23985 non-null  float64
 12  WindSpeed3pm   23327 non-null  float64
 13  Humidity9am    24171 non-null  float64
 14  Humidity3pm    23498 non-null  float64
 15  Pressure9am    20172 non-null  float64
 16  Pressure3pm    20173 non-null  float64
 17  Cloud9am       14136 non-null  float64
 18  Cloud3pm       13815 non-null  float64
 19  Temp9am        24312 non-null  float64
 20  Temp3pm        23639 non-null  float64
 21  RainToday      24280 non-null  object
 22  RISK_MM        24557 non-null  float64
 23  RainTomorrow   24557 non-null  object
dtypes: float64(17), object(7)
memory usage: 4.5+ MB
```

[53]:
```
fig, axes = plt.subplots(nrows=6, ncols=3, figsize=(12, 18))
axes = axes.reshape(-1)

continuous = [col for col in df.columns if df[col].dtype != object]
for i, col in enumerate(continuous):
    sns.histplot(df[col], ax=axes[i])

fig.tight_layout(pad=2.0)
plt.title('Histograms of Columns')
```

[53]:
```
Text(0.5, 1.0, 'Histograms of Columns')
```


Count
MinTemp
MaxTemp
Rainfall
Evaporation
Sunshine
WindGustSpeed
WindSpeed9am
WindSpeed3pm
Humidity9am
Humidity3pm
Pressure9am
Pressure3pm
Cloud9am
Cloud3pm
Temp9am
Temp3pm
RISK_MM
Histograms of Columns

Comment : - Except 'Date' and 'Location', other columns are missing values

[54]:
```
'Label encoder transform String column to Int type'

from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()

continuous = [col for col in df.columns if df[col].dtype == object]
continuous.remove('Date')

for i in continuous :
    df['{}'.format(i)] = le.fit_transform(df['{}'.format(i)])

df
```

[54]:

	Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation	\
0	2008-12-01	0	13.4	22.9	0.6	NaN	
1	2008-12-02	0	7.4	25.1	0.0	NaN	
2	2008-12-03	0	12.9	25.7	0.0	NaN	
3	2008-12-04	0	9.2	28.0	0.0	NaN	
4	2008-12-05	0	17.5	32.3	1.0	NaN	
…	…	…	…	…	…	…	
24553	2011-10-23	8	15.1	30.7	0.0	NaN	
24554	2011-10-24	8	14.9	34.5	0.0	NaN	
24555	2011-10-25	8	18.9	19.9	0.4	NaN	
24556	2011-10-26	8	12.3	15.9	8.2	NaN	
24557	2011-10-27	8	12.3	17.0	NaN	NaN	

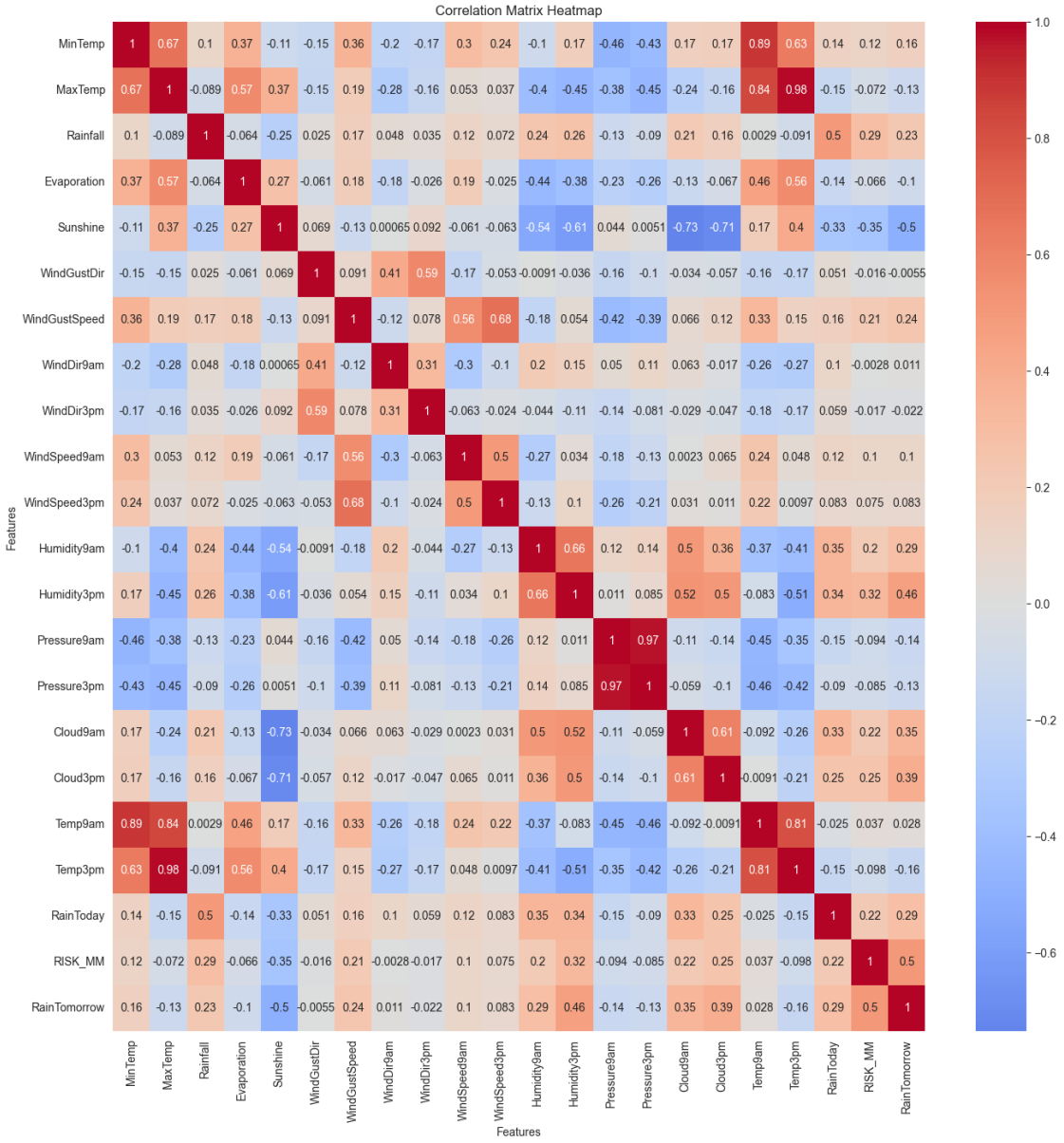
	Sunshine	WindGustDir	WindGustSpeed	WindDir9am	…	Humidity3pm	\
0	NaN	13	44.0	13	…	22.0	
1	NaN	14	44.0	6	…	25.0	
2	NaN	15	46.0	13	…	30.0	
3	NaN	4	24.0	9	…	16.0	
4	NaN	13	41.0	1	…	33.0	
…	…	…	…	…	…	…	
24553	NaN	5	20.0	3	…	31.0	
24554	NaN	7	39.0	14	…	25.0	
24555	NaN	8	43.0	9	…	98.0	
24556	NaN	9	31.0	12	…	74.0	
24557	NaN	16	NaN	16	…	NaN	

	Pressure9am	Pressure3pm	Cloud9am	Cloud3pm	Temp9am	Temp3pm	\
0	1007.7	1007.1	8.0	NaN	16.9	21.8	
1	1010.6	1007.8	NaN	NaN	17.2	24.3	
2	1007.6	1008.7	NaN	2.0	21.0	23.2	

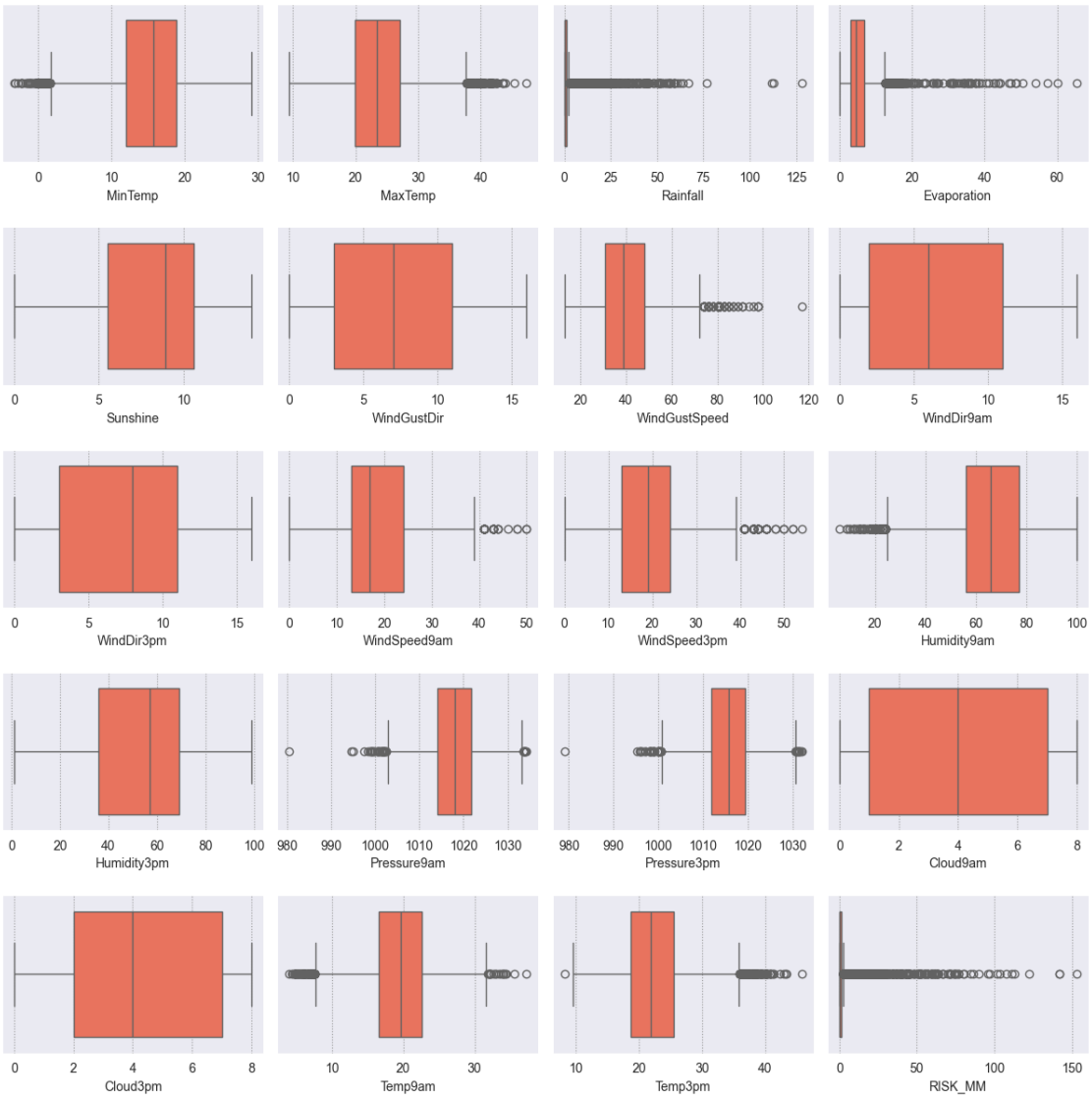
```
3           1017.6       1012.8       NaN       NaN     18.1     26.5
4           1010.8       1006.0       7.0       8.0     17.8     29.7
…             …            …          …         …       …        …
24553         NaN          NaN        NaN       NaN     20.6     30.1
24554         NaN          NaN        NaN       NaN     21.1     33.8
24555         NaN          NaN        NaN       NaN     19.0     16.1
24556         NaN          NaN        NaN       NaN     13.8     15.4
24557         NaN          NaN        NaN       NaN      NaN      NaN

       RainToday  RISK_MM  RainTomorrow
0              0      0.0             0
1              0      0.0             0
2              0      0.0             0
3              0      1.0             0
4              0      0.2             0
…            …      …             …
24553          0      0.0             0
24554          0      0.4             0
24555          0      8.2             1
24556          1      1.6             1
24557          2      NaN             2

[24558 rows x 24 columns]
```

- Comment : We can not draw histogram of 'Rainfall' and "RISK_MM" as the ourlier is large

[55]:
```
'Correlation matrix'
plt.figure(figsize=(16, 16))
df_ = df.drop (['Location', 'Date'], axis=1)
correlation  = df_.corr()

# Create heatmap
sns.heatmap(correlation, cmap='coolwarm',
            center=0, annot=True)

# Set title and axis labels
plt.title('Correlation Matrix Heatmap')
plt.xlabel('Features')
plt.ylabel('Features')
# plt.yticks(rotation = 30)

# Show plot
plt.show()
```



[67]:
```
fig = plt.figure(figsize=(12, 12))
temp = df.drop(['Location', 'Date', 'RainTomorrow', 'RainToday'], axis=1).
  ↪columns.tolist()
for i, item in enumerate(temp):
    plt.subplot(5, 4, i+1)
    sns.boxplot(data=df, x=item, color='tomato')
plt.tight_layout(pad=0.4, w_pad=0.5, h_pad=2.0)
plt.show()
```



[56]:
```
df = df.dropna()
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 6408 entries, 5939 to 23199
Data columns (total 24 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Date           6408 non-null   object
 1   Location       6408 non-null   int64
 2   MinTemp        6408 non-null   float64
 3   MaxTemp        6408 non-null   float64
 4   Rainfall       6408 non-null   float64
```

```
 5   Evaporation    6408 non-null   float64
 6   Sunshine       6408 non-null   float64
 7   WindGustDir    6408 non-null   int64
 8   WindGustSpeed  6408 non-null   float64
 9   WindDir9am     6408 non-null   int64
 10  WindDir3pm     6408 non-null   int64
 11  WindSpeed9am   6408 non-null   float64
 12  WindSpeed3pm   6408 non-null   float64
 13  Humidity9am    6408 non-null   float64
 14  Humidity3pm    6408 non-null   float64
 15  Pressure9am    6408 non-null   float64
 16  Pressure3pm    6408 non-null   float64
 17  Cloud9am       6408 non-null   float64
 18  Cloud3pm       6408 non-null   float64
 19  Temp9am        6408 non-null   float64
 20  Temp3pm        6408 non-null   float64
 21  RainToday      6408 non-null   int64
 22  RISK_MM        6408 non-null   float64
 23  RainTomorrow   6408 non-null   int64
dtypes: float64(17), int64(6), object(1)
memory usage: 1.2+ MB
```

[57]:
```
X = df.drop(['Date', 'RainTomorrow'], axis = 1)
y = df['RainTomorrow']

print (X.shape)
print (y.shape)
```

```
(6408, 22)
(6408,)
```

[58]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,␣
 ↪random_state=44)
```

[59]:
```
scaler = StandardScaler()

# Fit the StandardScaler on the training dataset
scaler.fit(X_train)

# Transform the training dataset
# using the StandardScaler
x_train_scaled = scaler.transform(X_train)
x_test_scaled = scaler.transform(X_test)
```

XGBoost

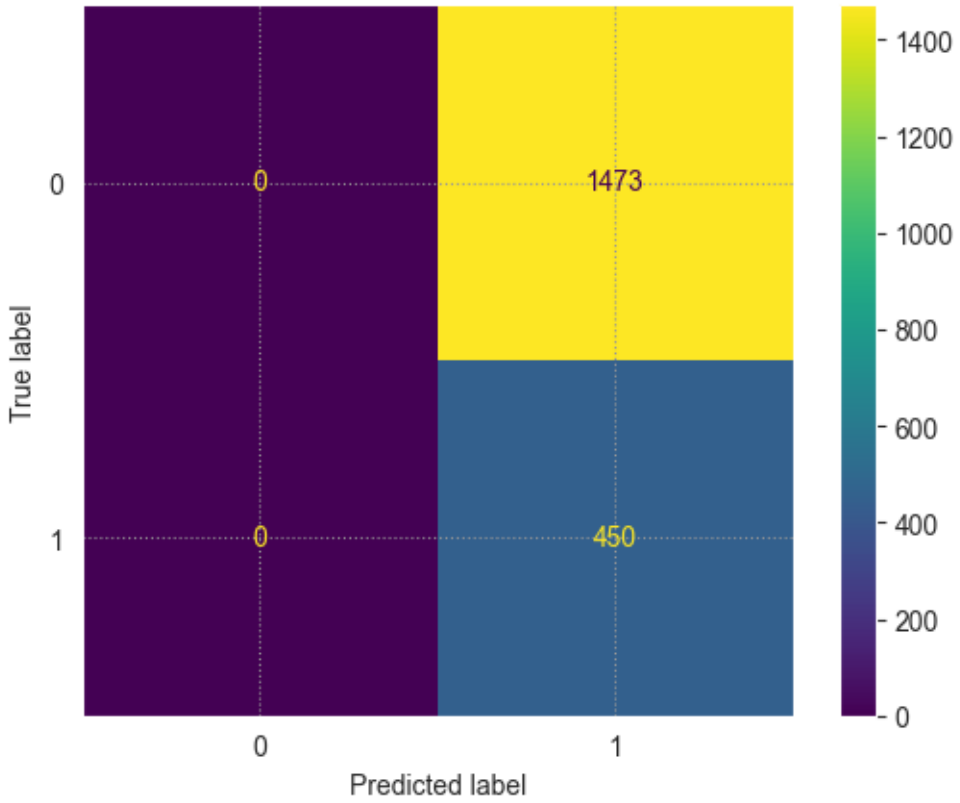
[60]:
```
# Create an instance of the XGBRegressor model
model_xgb = XGBClassifier()

# Fit the model to the training data
model_xgb.fit(x_train_scaled, y_train)


# Print the R-squared score on the training data
print("Xgboost Accuracy =", r2_score(
    y_train, model_xgb.predict(x_train_scaled)))

# Print the R-squared score on the test data
predictions = model_xgb.predict(x_test_scaled)
print("Xgboost Accuracy on test data =",
      r2_score(y_test, predictions))
```

```
Xgboost Accuracy = 1.0
Xgboost Accuracy on test data = 1.0
```

[72]:
```
print ('------ XGboost for un-standard data-----')
print("Xgboost Accuracy =", r2_score(
    y_train, model_xgb.predict(X_train)))

# Print the R-squared score on the test data
pred = model_xgb.predict(X_test)
print("Xgboost Accuracy on test data =",
      r2_score(y_test, pred))
```

```
------ XGboost for un-standard data-----
Xgboost Accuracy = -3.2073170731707314
Xgboost Accuracy on test data = -3.2733333333333334
```

[68]:
```
from sklearn.metrics import ConfusionMatrixDisplay,confusion_matrix,␣
 ↪classification_report

disp = ConfusionMatrixDisplay.from_predictions(y_test, predictions)
plt.show()
```



[70]:
```
print (classification_report(y_test, predictions))
```

```
              precision    recall  f1-score   support

           0       0.00      0.00      0.00      1473
           1       0.23      1.00      0.38       450

    accuracy                           0.23      1923
   macro avg       0.12      0.50      0.19      1923
weighted avg       0.05      0.23      0.09      1923
```