In [16]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
```

In [3]:
```
data = pd.read_csv(r'C:\Users\mayur/weatherAUS.csv')
data.head()
```

Out[3]:

	Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDir	WindGustSpeed
0	2008-12-01	Albury	13.4	22.9	0.6	NaN	NaN	W	44.0
1	2008-12-02	Albury	7.4	25.1	0.0	NaN	NaN	WNW	44.0
2	2008-12-03	Albury	12.9	25.7	0.0	NaN	NaN	WSW	46.0
3	2008-12-04	Albury	9.2	28.0	0.0	NaN	NaN	NE	24.0
4	2008-12-05	Albury	17.5	32.3	1.0	NaN	NaN	W	41.0

5 rows × 23 columns

In [4]:
```
data.describe()
```

Out[4]:

	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustSpeed	Wind
count	143975.000000	144199.000000	142199.000000	82670.000000	75625.000000	135197.000000	143
mean	12.194034	23.221348	2.360918	5.468232	7.611178	40.035230	
std	6.398495	7.119049	8.478060	4.193704	3.785483	13.607062	
min	-8.500000	-4.800000	0.000000	0.000000	0.000000	6.000000	
25%	7.600000	17.900000	0.000000	2.600000	4.800000	31.000000	
50%	12.000000	22.600000	0.000000	4.800000	8.400000	39.000000	
75%	16.900000	28.200000	0.800000	7.400000	10.600000	48.000000	
max	33.900000	48.100000	371.000000	145.000000	14.500000	135.000000	

In [5]:
```
data.shape
```

Out[5]:
```
(145460, 23)
```

In [6]:
```
data.isnull().sum()
```

Out[6]:
```
Date                0
Location            0
MinTemp          1485
MaxTemp          1261
Rainfall         3261
Evaporation     62790
Sunshine        69835
WindGustDir     10326
WindGustSpeed   10263
WindDir9am      10566
WindDir3pm       4228
WindSpeed9am     1767
WindSpeed3pm     3062
Humidity9am      2654
Humidity3pm      4507
Pressure9am     15065
Pressure3pm     15028
Cloud9am        55888
Cloud3pm        59358
Temp9am          1767
Temp3pm          3609
RainToday        3261
RainTomorrow     3267
dtype: int64
```

In [7]:
```
data = data.drop(["Evaporation","Sunshine","Cloud9am","Cloud3pm","Location", "Date"], a
data.head()
```

Out[7]:

	MinTemp	MaxTemp	Rainfall	WindGustDir	WindGustSpeed	WindDir9am	WindDir3pm	WindSpeed
0	13.4	22.9	0.6	W	44.0	W	WNW	
1	7.4	25.1	0.0	WNW	44.0	NNW	WSW	
2	12.9	25.7	0.0	WSW	46.0	W	WSW	
3	9.2	28.0	0.0	NE	24.0	SE	E	
4	17.5	32.3	1.0	W	41.0	ENE	NW	

In [8]:
```
data = data.dropna(axis = 0)
data.shape
```

Out[8]:
```
(112925, 17)
```

In [9]:
```
data.columns
```

Out[9]:
```
Index(['MinTemp', 'MaxTemp', 'Rainfall', 'WindGustDir', 'WindGustSpeed',
       'WindDir9am', 'WindDir3pm', 'WindSpeed9am', 'WindSpeed3pm',
       'Humidity9am', 'Humidity3pm', 'Pressure9am', 'Pressure3pm', 'Temp9am',
       'Temp3pm', 'RainToday', 'RainTomorrow'],
      dtype='object')
```

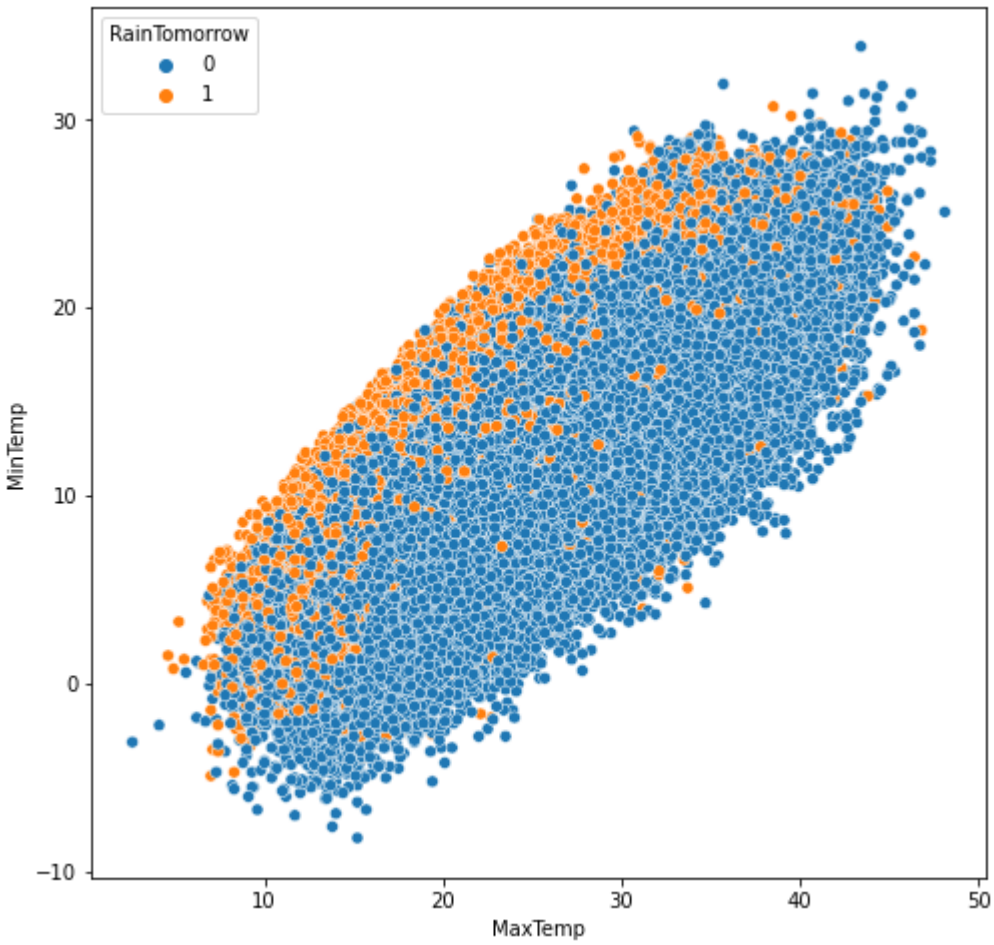
In [11]:
```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
```

```
data['WindGustDir'] = le.fit_transform(data['WindGustDir'])
data['WindDir9am'] = le.fit_transform(data['WindDir9am'])
data['WindDir3pm'] = le.fit_transform(data['WindDir3pm'])
data['RainToday'] = le.fit_transform(data['RainToday'])
data['RainTomorrow'] = le.fit_transform(data['RainTomorrow'])
```

In [12]:
```
x = data.drop(['RainTomorrow'], axis = 1)
y = data['RainTomorrow']
```

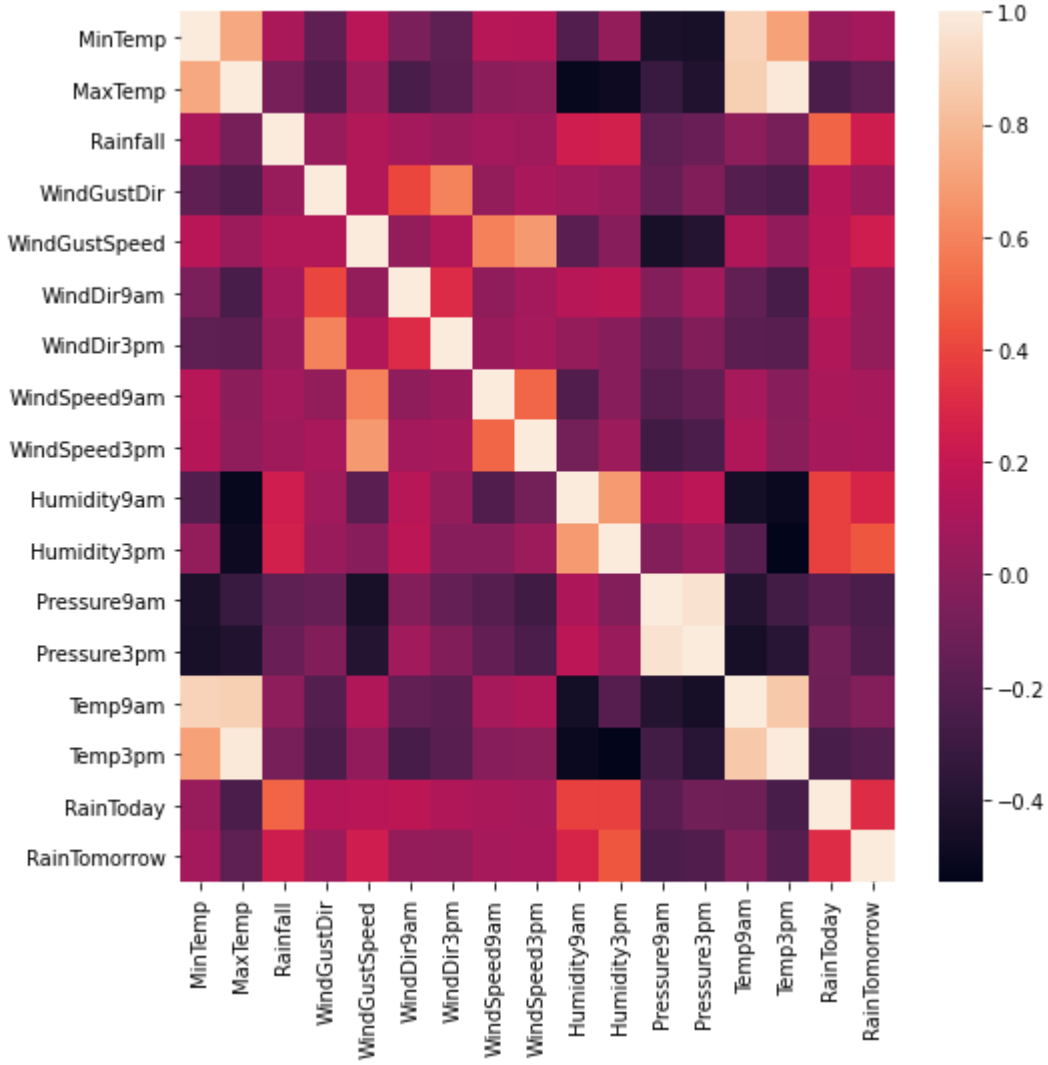
In [13]:
```
plt.figure(figsize = (8,8))
sns.scatterplot(x = 'MaxTemp', y = 'MinTemp', hue = 'RainTomorrow' ,data = data)
```

Out[13]: <AxesSubplot:xlabel='MaxTemp', ylabel='MinTemp'>



In [14]:
```
plt.figure(figsize = (8,8))
sns.heatmap(data.corr())
```

Out[14]: <AxesSubplot:>



In [15]:
```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x,y, test_size = 0.2,random_state=1
```

In [17]:
```
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression()
lr.fit(x_train,y_train)
predictions = lr.predict(x_test)
print(confusion_matrix(y_test, predictions))
print(classification_report(y_test, predictions))
print(accuracy_score(y_test, predictions))
```

```
[[16668   852]
 [ 2630  2435]]
              precision    recall  f1-score   support

           0       0.86      0.95      0.91     17520
           1       0.74      0.48      0.58      5065

    accuracy                           0.85     22585
   macro avg       0.80      0.72      0.74     22585
weighted avg       0.84      0.85      0.83     22585

0.8458268762452955
C:\Users\mayur\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:763: Conver
```

```
genceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

In [ ]: