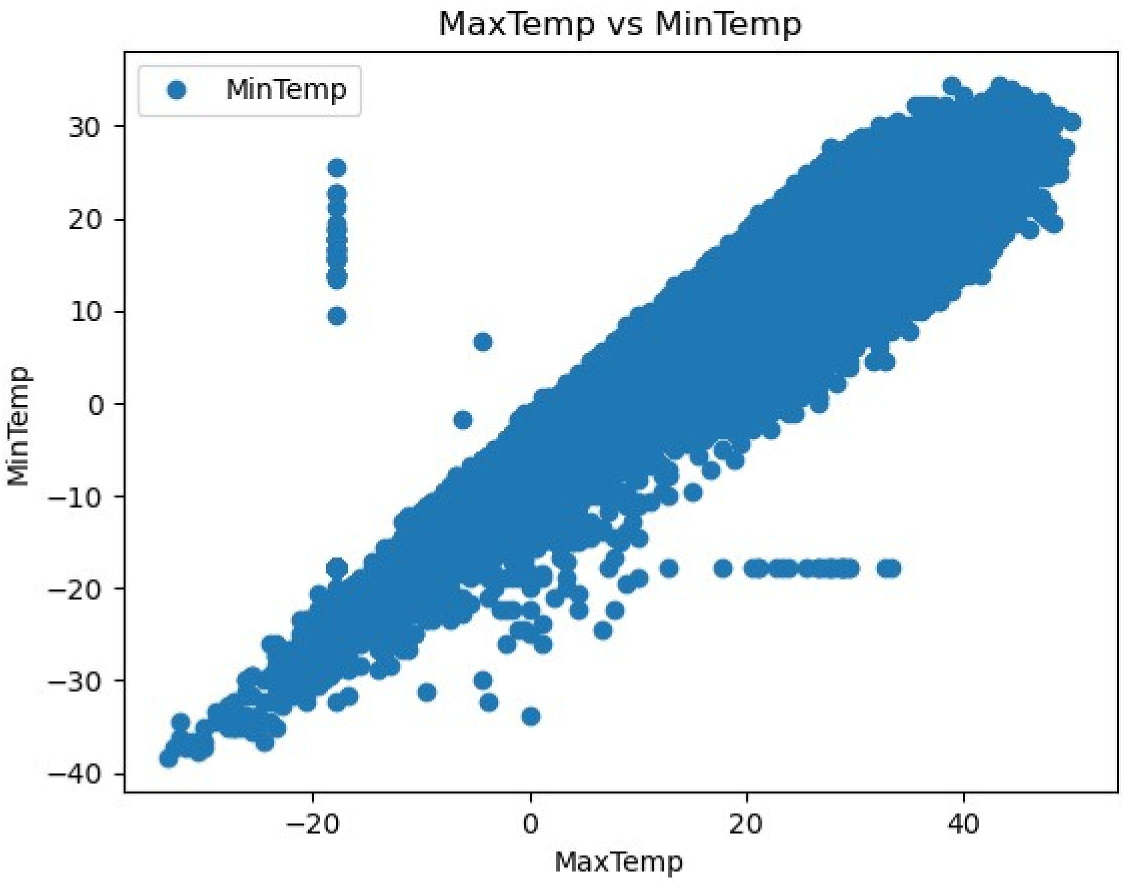
TASK 2:

Apply Linear Regression on the dataset provided with the manual (weather.xlsx) and predict the minimum temperature based on the maximum temperature. Elaborate the results.

In [16]:

```
import numpy as np
import matplotlib.pyplot as plot
import pandas
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn import metrics

dataset = pd.read_csv('Weather.csv')

x = dataset['MaxTemp'].values.reshape(-1, 1)
y = dataset['MinTemp'].values.reshape(-1, 1)

dataset.plot(x='MaxTemp', y='MinTemp', style='o')
plot.title('MaxTemp vs MinTemp')
plot.xlabel('MaxTemp')
plot.ylabel('MinTemp')
plot.show()

xTrain, xTest, yTrain, yTest = train_test_split(x, y, test_size=0.2, random_st

linearRegressor = LinearRegression()
linearRegressor.fit(xTrain, yTrain)

yPrediction = linearRegressor.predict(xTest)


print('Mean Absolute Error:', metrics.mean_absolute_error(yTest, yPrediction))
print('Mean Squared Error:', metrics.mean_squared_error(yTest, yPrediction))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(yTest, yP

df = pandas.DataFrame({'Actual': yTest.flatten(), 'Predicted': predict_test.fl
df
```

```
C:\Users\COMPUTER POINT\AppData\Local\Temp\ipykernel_3048\2821346738.py:8: Dt
ypeWarning: Columns (7,8,18,25) have mixed types. Specify dtype option on imp
ort or set low_memory=False.
  dataset = pd.read_csv('Weather.csv')
```

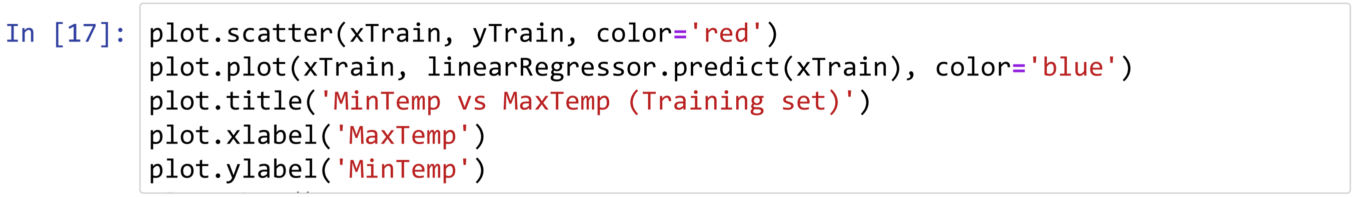


```
Mean Absolute Error: 3.1367056440832815
Mean Squared Error: 16.237989240292094
Root Mean Squared Error: 4.02963884737728
```

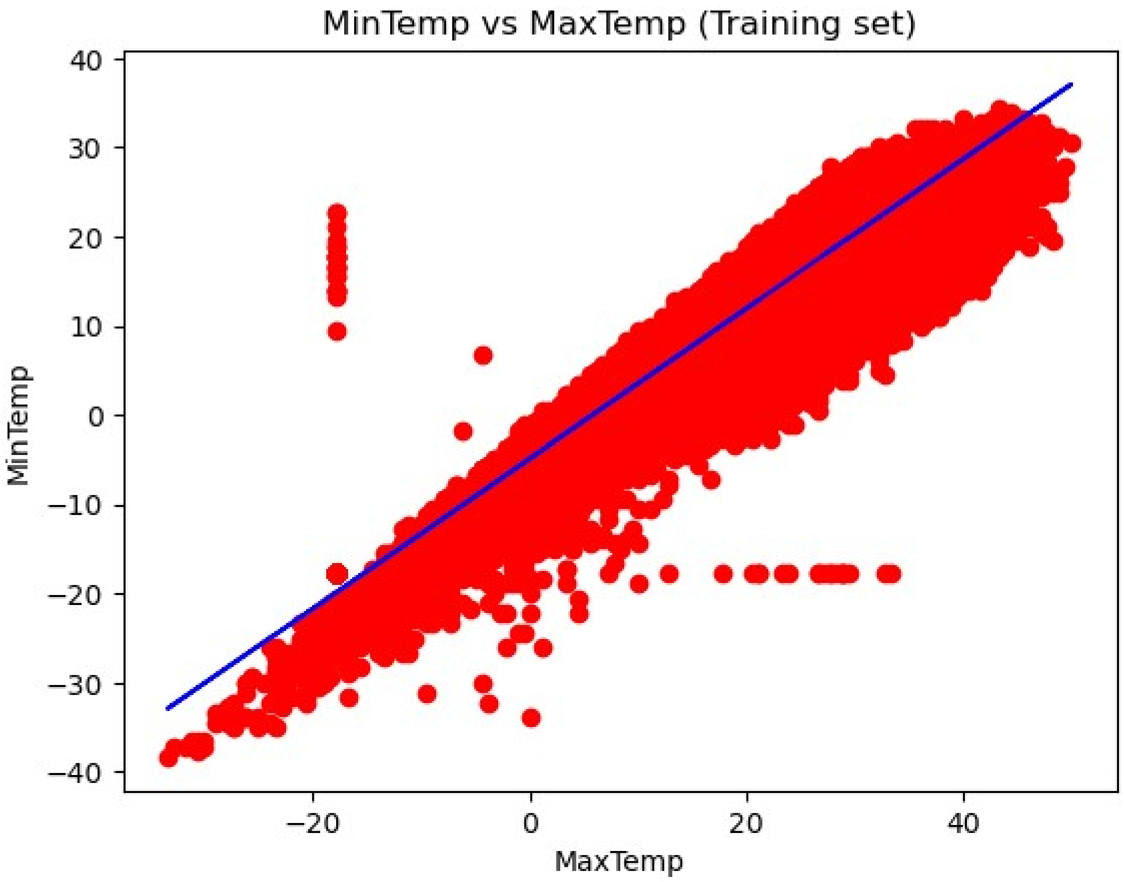
Out[16]:

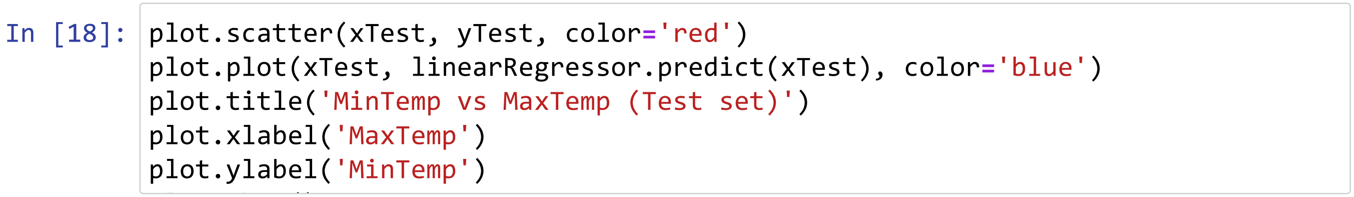
	Actual	Predicted
0	25.000000	21.427747
1	21.111111	21.746762
2	17.222222	22.000672
3	22.222222	21.681174
4	5.555556	21.789182
...	...	...
23803	23.333333	-18.963088
23804	20.000000	21.943528
23805	23.888889	3.529173
23806	21.666667	21.812351
23807	22.777778	13.328365

23808 rows × 2 columns

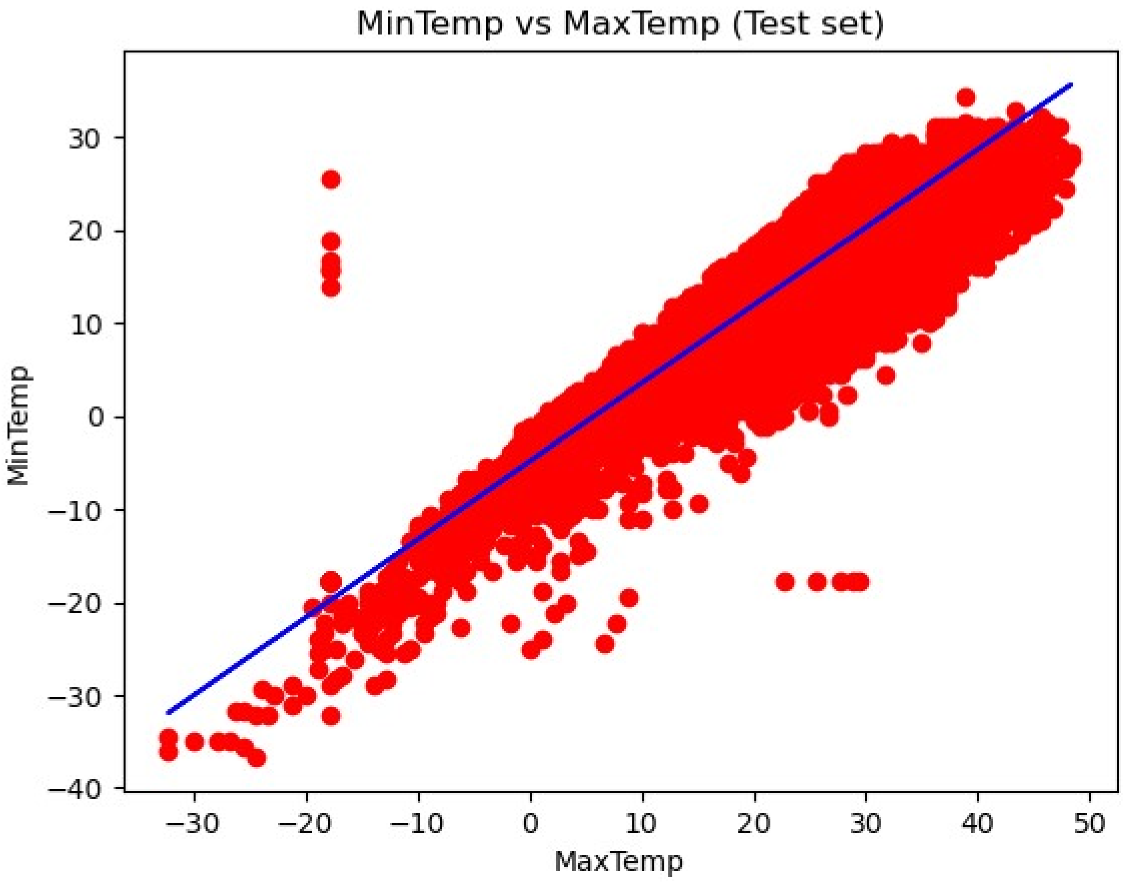


```
In [17]: plot.scatter(xTrain, yTrain, color='red')
         plot.plot(xTrain, linearRegressor.predict(xTrain), color='blue')
         plot.title('MinTemp vs MaxTemp (Training set)')
         plot.xlabel('MaxTemp')
         plot.ylabel('MinTemp')
```





```
In [18]: plot.scatter(xTest, yTest, color='red')
         plot.plot(xTest, linearRegressor.predict(xTest), color='blue')
         plot.title('MinTemp vs MaxTemp (Test set)')
         plot.xlabel('MaxTemp')
         plot.ylabel('MinTemp')
```



TASK 3:

Apply a Neural Network on the dataset provided with the manual (weather.xlsx) and predict the minimum temperature based on the maximum temperature. Elaborate the results.

In [22]:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPRegressor
from sklearn import metrics

dataset = pd.read_csv('Weather.csv')

xTrain, xTest, yTrain, yTest = train_test_split(x, y, test_size=0.2, random_st

mlp_R = MLPRegressor(hidden_layer_sizes=(18, 18, 18), activation='relu', solve
mlp_R.fit(xTrain, yTrain.ravel())

predict_test = mlp_R.predict(xTest)

print(predict_test)

print('Mean Absolute Error:', metrics.mean_absolute_error(yTest, predict_test)
print('Mean Squared Error:', metrics.mean_squared_error(yTest, predict_test))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(yTest, pr

df = pandas.DataFrame({'Actual': yTest.flatten(), 'Predicted': predict_test.fl
df
```

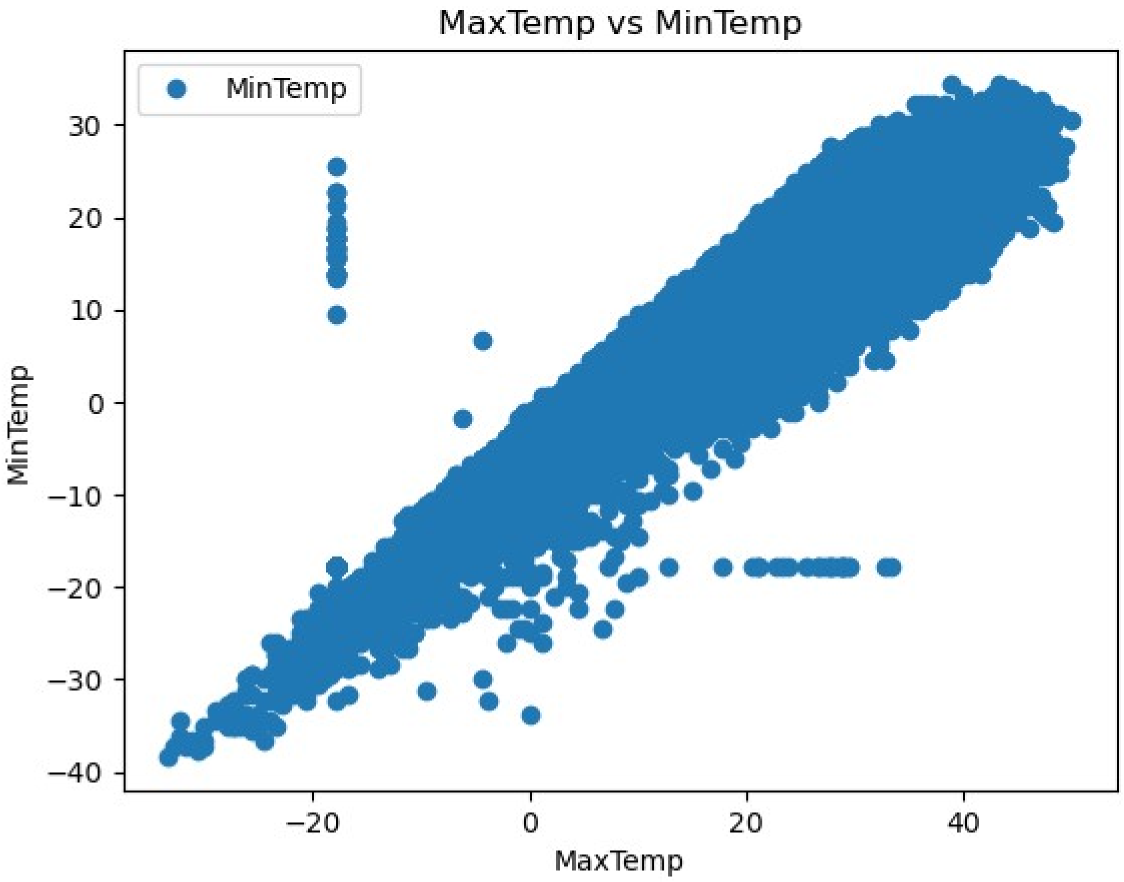
```
C:\Users\COMPUTER POINT\AppData\Local\Temp\ipykernel_3048\3481822250.py:8: Dt
ypeWarning: Columns (7,8,18,25) have mixed types. Specify dtype option on imp
ort or set low_memory=False.
  dataset = pd.read_csv('Weather.csv')

[21.3187044  21.47417267 22.31140599 ...  3.60658392 21.55266831
 13.02118279]
Mean Absolute Error: 2.7863663701699477
Mean Squared Error: 13.34961815902122
Root Mean Squared Error: 3.6537129278339893
```

Out[22]:

	Actual	Predicted
0	11.111111	21.318704
1	19.444444	21.474173
2	22.777778	22.311406
3	21.666667	21.395677
4	23.333333	21.722788
...	...	...
23803	-18.888889	-20.423452
23804	21.111111	21.709660
23805	5.555556	3.606584
23806	22.222222	21.552668
23807	15.000000	13.021183

23808 rows × 2 columns

In [23]:
```
x = dataset['MaxTemp'].values.reshape(-1, 1)
y = dataset['MinTemp'].values.reshape(-1, 1)

dataset.plot(x='MaxTemp', y='MinTemp', style='o')
plt.title('MaxTemp vs MinTemp')
plt.xlabel('MaxTemp')
plt.ylabel('MinTemp')
plt.show()
```


MaxTemp vs MinTemp
● MinTemp
MinTemp
MaxTemp