```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/AirQualityUCI.csv')
df
```

	Date	Time	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)	PT08.S4(NO
0	10/03/2004	18:00:00	2.6	1360.0	150.0	11.9	1046.0	166.0	1056.0	113.0	169
1	10/03/2004	19:00:00	2.0	1292.0	112.0	9.4	955.0	103.0	1174.0	92.0	155
2	10/03/2004	20:00:00	2.2	1402.0	88.0	9.0	939.0	131.0	1140.0	114.0	155
3	10/03/2004	21:00:00	2.2	1376.0	80.0	9.2	948.0	172.0	1092.0	122.0	158
4	10/03/2004	22:00:00	1.6	1272.0	51.0	6.5	836.0	131.0	1205.0	116.0	149
...	...	...	...	...	...	...	...	...	...	...	
9466	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
9467	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
9468	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
[illegible]	[illegible]	[illegible]	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
[illegible]	[illegible]	[illegible]	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N

Saved successfully! ×

9471 rows × 17 columns

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9471 entries, 0 to 9470
Data columns (total 17 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Date           9357 non-null   object 
 1   Time           9357 non-null   object 
 2   CO(GT)         9357 non-null   float64
 3   PT08.S1(CO)    9357 non-null   float64
 4   NMHC(GT)       9357 non-null   float64
 5   C6H6(GT)       9357 non-null   float64
 6   PT08.S2(NMHC)  9357 non-null   float64
 7   NOx(GT)        9357 non-null   float64
 8   PT08.S3(NOx)   9357 non-null   float64
 9   NO2(GT)        9357 non-null   float64
 10  PT08.S4(NO2)   9357 non-null   float64
 11  PT08.S5(O3)    9357 non-null   float64
 12  T              9357 non-null   float64
 13  RH             9357 non-null   float64
 14  AH             9357 non-null   float64
 15  Unnamed: 15    0 non-null      float64
 16  Unnamed: 16    0 non-null      float64
dtypes: float64(15), object(2)
memory usage: 1.2+ MB
```

```
df.dropna(subset = ['AH'], axis = 0, inplace = True)
```

```
df.reset_index(drop = True, inplace = True)
```

```
df.drop(['Unnamed: 15', 'Unnamed: 16'], axis = 1, inplace = True)
```

```
df.tail(5)
```

	Date	Time	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)	PT08.S4(NO
9352	04/04/2005	10:00:00	3.1	1314.0	-200.0	13.5	1101.0	472.0	539.0	190.0	137
9353	04/04/2005	11:00:00	2.4	1163.0	-200.0	11.4	1027.0	353.0	604.0	179.0	126

```
df['Date'] = df['Date'].astype('category')
df['Date'] = df['Date'].cat.codes

df['Time'] = df['Time'].astype('category')
df['Time'] = df['Time'].cat.codes
```

```
df.head(5)
```

	Date	Time	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)	PT08.S4(NO2)	PT08.S5
0	114	18	2.6	1360.0	150.0	11.9	1046.0	166.0	1056.0	113.0	1692.0	12
1	114	19	2.0	1292.0	112.0	9.4	955.0	103.0	1174.0	92.0	1559.0	9
2	114	20	2.2	1402.0	88.0	9.0	939.0	131.0	1140.0	114.0	1555.0	10
3	114	21	2.2	1376.0	80.0	9.2	948.0	172.0	1092.0	122.0	1584.0	12
4	114	22	1.6	1272.0	51.0	6.5	836.0	131.0	1205.0	116.0	1490.0	11

Saved successfully! ×

	Date	Time	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)	PT08.S4(NO2)	PT08
0	114	18	2.6	1360.0	150.0	11.9	1046.0	166.0	1056.0	113.0	1692.0	
1	114	19	2.0	1292.0	112.0	9.4	955.0	103.0	1174.0	92.0	1559.0	
2	114	20	2.2	1402.0	88.0	9.0	939.0	131.0	1140.0	114.0	1555.0	
3	114	21	2.2	1376.0	80.0	9.2	948.0	172.0	1092.0	122.0	1584.0	
4	114	22	1.6	1272.0	51.0	6.5	836.0	131.0	1205.0	116.0	1490.0	
...	...	...	...	...	...	...	...	...	...	...	...	
9352	43	10	3.1	1314.0	-200.0	13.5	1101.0	472.0	539.0	190.0	1374.0	
9353	43	11	2.4	1163.0	-200.0	11.4	1027.0	353.0	604.0	179.0	1264.0	
9354	43	12	2.4	1142.0	-200.0	12.4	1063.0	293.0	603.0	175.0	1241.0	
9355	43	13	2.1	1003.0	-200.0	9.5	961.0	235.0	702.0	156.0	1041.0	
9356	43	14	2.2	1071.0	-200.0	11.9	1047.0	265.0	654.0	168.0	1129.0	

9357 rows × 14 columns

```
y = df['AH']
```

```
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 = 0)
```

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_train_scaler = scaler.fit_transform(X_train)
X_test_scaler = scaler.transform(X_test)
```

```
from sklearn.tree import DecisionTreeRegressor
reg = DecisionTreeRegressor(max_depth = 10, min_samples_split = 10)
```

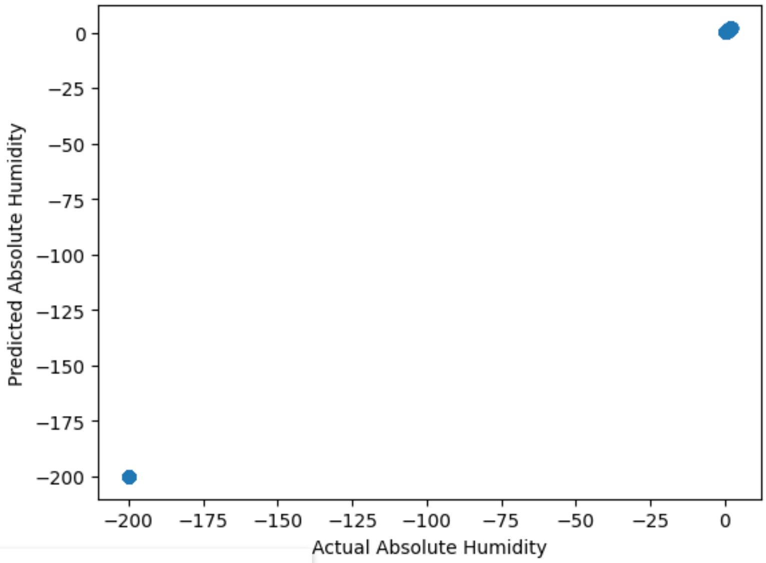
```
reg.fit(X_train_scaler, y_train)
```

```
▾              DecisionTreeRegressor
DecisionTreeRegressor(max_depth=10, min_samples_split=10)
```

```
y_pred_train = reg.predict(X_train_scaler)
y_pred_train
```

```
array([0.50248571, 1.76207692, 1.53176923, ..., 1.09208421, 1.6012    ,
       1.28659189])
```

```
plt.scatter(y_train, y_pred_train)
plt.xlabel("Actual Absolute Humidity")
plt.ylabel("Predicted Absolute Humidity")
plt.show()
```



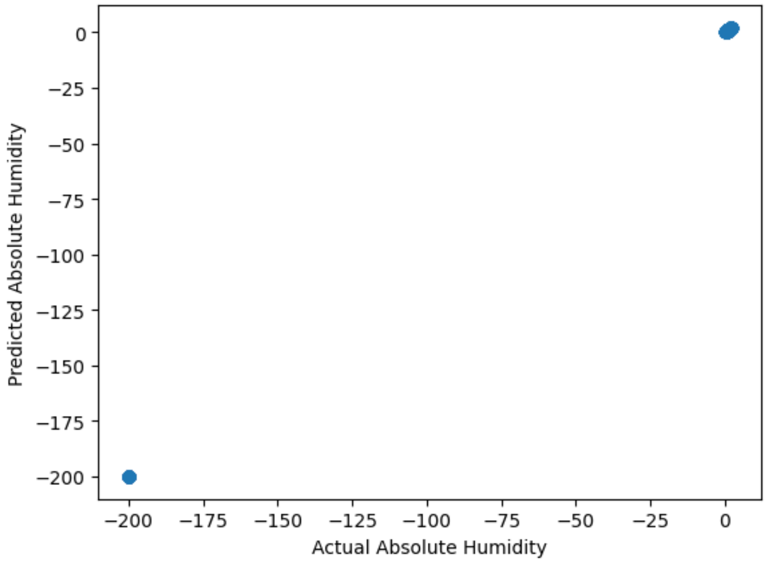
Saved successfully! ×

```
from sklearn.metrics import r2_score
r2_score(y_train, y_pred_train)
```

```
    0.9999993194224269
```

```
y_pred_test = reg.predict(X_test_scaler)
```

```
plt.scatter(y_test, y_pred_test)
plt.xlabel("Actual Absolute Humidity")
plt.ylabel("Predicted Absolute Humidity")
plt.show()
```



```
r2_score(y_test, y_pred_test)
```

```
    0.9999987285556645
```

✓ 0s completed at 19:34

Saved successfully!