Weather prediction

November 23, 2022

```
[40]: import pandas as pd
      import numpy as np
      import plotly.express as px
      import plotly.graph_objects as go
      from matplotlib import pyplot as plt
```

```
[3]: data = pd.read_csv("data.csv")
```

```
[5]: data.head(10)
```

```
[5]:    Unnamed: 0  YEAR    JAN    FEB    MAR    APR    MAY    JUN    JUL    AUG  \
     0           0  1901  17.99  19.43  23.49  26.41  28.28  28.60  27.49  26.98
     1           1  1902  19.00  20.39  24.10  26.54  28.68  28.44  27.29  27.05
     2           2  1903  18.32  19.79  22.46  26.03  27.93  28.41  28.04  26.63
     3           3  1904  17.77  19.39  22.95  26.73  27.83  27.85  26.84  26.73
     4           4  1905  17.40  17.79  21.78  24.84  28.32  28.69  27.67  27.47
     5           5  1906  17.50  19.14  22.21  26.53  29.06  28.02  27.46  26.82
     6           6  1907  19.27  19.42  22.03  26.98  27.52  27.66  27.28  26.38
     7           7  1908  18.35  19.73  22.93  27.06  28.07  28.49  27.16  26.41
     8           8  1909  17.79  19.05  23.40  25.76  27.97  27.67  26.56  26.43
     9           9  1910  18.14  19.72  22.90  25.96  28.36  27.72  26.93  26.61

          SEP    OCT    NOV    DEC
     0  26.26  25.08  21.73  18.95
     1  25.95  24.37  21.33  18.78
     2  26.34  24.57  20.96  18.29
     3  25.84  24.36  21.07  18.84
     4  26.29  26.16  22.07  18.71
     5  26.23  24.75  21.93  19.55
     6  26.26  24.72  22.11  18.46
     7  25.74  24.25  21.06  18.15
     8  25.47  24.37  22.01  18.86
     9  25.98  24.04  20.72  18.05
```

```
[11]: data.tail()
```

```
[11]:      Unnamed: 0  YEAR    JAN    FEB    MAR    APR    MAY    JUN    JUL    AUG  \
      112         112  2013  18.88  21.07  24.53  26.97  29.06  28.24  27.50  27.22
```

```
113         113  2014  18.81  20.35  23.34  26.91  28.45  29.42  28.07  27.42
114         114  2015  19.02  21.23  23.52  26.52  28.82  28.15  28.03  27.64
115         115  2016  20.92  23.58  26.61  29.56  30.41  29.70  28.18  28.17
116         116  2017  20.59  23.08  25.58  29.17  30.47  29.44  28.31  28.12

       SEP    OCT    NOV    DEC
112  26.87  25.63  22.18  19.69
113  26.61  25.38  22.53  19.50
114  27.04  25.82  22.95  20.21
115  27.72  26.81  23.90  21.89
116  28.11  27.24  23.92  21.47
```

[9]:
```
data.apply(pd.isnull)
```

[9]:
```
     Unnamed: 0   YEAR    JAN    FEB    MAR    APR    MAY    JUN    JUL  \
0         False  False  False  False  False  False  False  False  False
1         False  False  False  False  False  False  False  False  False
2         False  False  False  False  False  False  False  False  False
3         False  False  False  False  False  False  False  False  False
4         False  False  False  False  False  False  False  False  False
..          ...    ...    ...    ...    ...    ...    ...    ...    ...
112       False  False  False  False  False  False  False  False  False
113       False  False  False  False  False  False  False  False  False
114       False  False  False  False  False  False  False  False  False
115       False  False  False  False  False  False  False  False  False
116       False  False  False  False  False  False  False  False  False

       AUG    SEP    OCT    NOV    DEC
0    False  False  False  False  False
1    False  False  False  False  False
2    False  False  False  False  False
3    False  False  False  False  False
4    False  False  False  False  False
..     ...    ...    ...    ...    ...
112  False  False  False  False  False
113  False  False  False  False  False
114  False  False  False  False  False
115  False  False  False  False  False
116  False  False  False  False  False

[117 rows x 14 columns]
```

[10]:
```
data.apply(pd.isnull).sum()
```

[10]:
```
Unnamed: 0    0
YEAR          0
JAN           0
```

```
FEB          0
MAR          0
APR          0
MAY          0
JUN          0
JUL          0
AUG          0
SEP          0
OCT          0
NOV          0
DEC          0
dtype: int64
```

```
[14]: data1 = pd.melt(data, id_vars = 'YEAR', value_vars = data.columns[1:])
```

```
[15]: data1.head()
```

```
[15]:    YEAR variable  value
      0  1901      JAN  17.99
      1  1902      JAN  19.00
      2  1903      JAN  18.32
      3  1904      JAN  17.77
      4  1905      JAN  17.40
```

```
[16]: data1['Date'] = data1['variable']+" "+ data1['YEAR'].astype(str)
```

```
[20]: from datetime import datetime
      from dateutil import parser
```

```
[25]: data1.loc[:,'Date'] = data1['Date'].apply(lambda x: datetime.strptime(x, '%b␣
        ↪%Y'))
```

```
[26]: data1.head()
```

```
[26]:    YEAR variable  value       Date
      0  1901      JAN  17.99 1901-01-01
      1  1902      JAN  19.00 1902-01-01
      2  1903      JAN  18.32 1903-01-01
      3  1904      JAN  17.77 1904-01-01
      4  1905      JAN  17.40 1905-01-01
```

```
[33]: data1.columns = ['Year','Month','Temp','Date']
      data1.sort_values(by = 'Date', inplace = True)
```

```
[43]: fig = go.Figure(layout = go.Layout(yaxis=dict(range=[0, data1['Temp'].
        ↪max()+1])))
```

[47]:

```
fig.add_trace(go.Scatter(x=data1['Date'], y=data1['Temp']), )
fig.update_layout(title='Temprature Throught Timeline:',
                  xaxis_title='Time', yaxis_title='Temprature in Degrees')
```

[50]:

```
fig = px.box(data1, 'Month', 'Temp')
fig.update_layout(title='Warmest, Coldest and Median Monthly Tempratue.')
fig.show()
```

[55]:

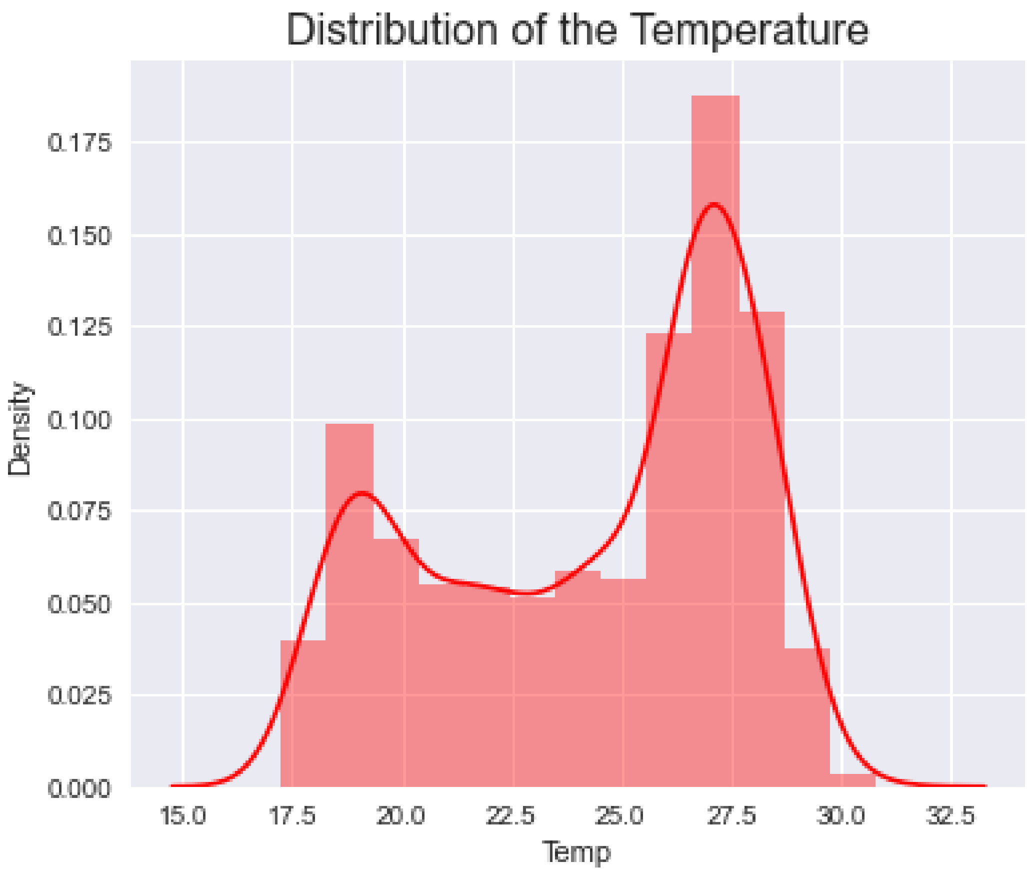
```
import seaborn as sns
```

[58]:

```
plt.figure(figsize=(6,5))
sns.distplot(data1['Temp'],color='red')
plt.title('Distribution of the Temperature', fontsize=16)
plt.show()
```

```
C:\Users\Sushan Shivagiri\AppData\Local\Programs\Python\Python310\lib\site-
packages\seaborn\distributions.py:2619: FutureWarning:

`distplot` is a deprecated function and will be removed in a future version.
Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).
```



[59]:

```python
fig = px.line(data1, 'Year', 'Temp', facet_col='Month', facet_col_wrap=4)
fig.update_layout(title='Monthly temprature throught history:')
fig.show()
```

[60]:

```python
from sklearn.tree import DecisionTreeRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score
```

[61]:

```python
data2 = data1[['Year', 'Month', 'Temp']].copy()
data2 = pd.get_dummies(data2)
```

[63]:

```python
y = data2[['Temp']]
x = data2.drop(columns='Temp')
```

[64]:

```python
dtr = DecisionTreeRegressor()
```

[65]:

```python
train_x, test_x, train_y, test_y = train_test_split(x,y,test_size=0.3)
```

[66]:

```python
dtr.fit(train_x, train_y)
pred = dtr.predict(test_x)
```

[67]:

```python
score = r2_score(test_y, pred)
```

[69]:

```python
print("Accuracy:", score*100)
```

```
Accuracy: 95.84499280677107
```