Assignment 11(Group B3)

Name : Mohit Manish Bhavsar

Roll No : 20U437

Div : 4

Problem Statement:

Locate dataset (e.g., sample_weather.txt) for working on weather data which reads the text input files and finds average for temperature, dew point and wind speed.

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
```

```
df = pd.read_csv("training_data_with_weather_info_week_1.csv")
```

```
df.head()
```

```
   Id Province/State Country/Region   Lat  Long        Date  ConfirmedCases
\
0   1            NaN    Afghanistan  33.0  65.0  2020-01-22             0.0
1   2            NaN    Afghanistan  33.0  65.0  2020-01-23             0.0
2   3            NaN    Afghanistan  33.0  65.0  2020-01-24             0.0
3   4            NaN    Afghanistan  33.0  65.0  2020-01-25             0.0
4   5            NaN    Afghanistan  33.0  65.0  2020-01-26             0.0

   Fatalities  day_from_jan_first  temp   min   max    stp     slp  dewp  \
0         0.0                  22  42.6  33.6  54.9  999.9  1024.3  27.4
1         0.0                  23  42.0  32.7  55.9  999.9  1020.8  22.8
2         0.0                  24  40.1  36.9  43.2  999.9  1018.6  34.5
3         0.0                  25  46.0  37.9  56.3  999.9  1018.0  37.8
4         0.0                  26  42.8  36.1  53.1  999.9  1014.8  33.2

         rh        ah  wdsp   prcp  fog
0  0.545709  0.186448   9.4   0.00    0
1  0.461259  0.163225  14.9  99.99    1
2  0.801794  0.325375  10.4   0.17    1
3  0.728175  0.214562   6.1   0.57    1
4  0.685513  0.231656  10.8   0.00    1
```

```
df.tail()
```

```
          Id Province/State Country/Region      Lat     Long        Date  \
17887  26378            NaN         Zambia -15.4167  28.2833  2020-03-20
17888  26379            NaN         Zambia -15.4167  28.2833  2020-03-21
17889  26380            NaN         Zambia -15.4167  28.2833  2020-03-22
17890  26381            NaN         Zambia -15.4167  28.2833  2020-03-23
17891  26382            NaN         Zambia -15.4167  28.2833  2020-03-24

       ConfirmedCases  Fatalities  day_from_jan_first  temp   min   max  \
17887             2.0         0.0                  80  70.6  62.6  81.9
17888             2.0         0.0                  81  71.3  66.2  81.5
17889             3.0         0.0                  82  72.1  67.1  80.4
17890             3.0         0.0                  83  71.7  66.2  80.6
17891             3.0         0.0                  84  72.6  60.3  84.2

         stp  slp  dewp        rh        ah  wdsp   prcp  fog
17887  999.9  NaN  62.8  0.761545  0.198068   6.0   0.00    0
17888  999.9  NaN  65.3  0.812047  0.212487   7.1  99.99    1
17889  999.9  NaN  66.7  0.829815  0.218712   5.0  99.99    1
17890  999.9  NaN  62.8  0.733343  0.192580   4.2   0.00    0
17891  999.9  NaN  62.0  0.691204  0.183033   6.4   0.00    1
```

```
df.shape
```

```
(17892, 20)
```

```
df.isnull().sum()
```

```
Id                      0
Province/State       9702
Country/Region          0
Lat                     0
Long                    0
Date                    0
ConfirmedCases          0
Fatalities              0
day_from_jan_first      0
temp                    0
min                   137
max                    16
stp                     0
slp                  6947
dewp                  618
rh                    618
ah                    618
wdsp                    0
prcp                    0
fog                     0
dtype: int64
```

```
mean_dew = df['dewp'].mean()
medain_dew = df["dewp"].median()
print("mean",mean_dew)
print("median",medain_dew)
```

```
mean 42.35362973254584
median 40.8
```

```
df.describe(include="all")
```

```
                  Id    Province/State Country/Region           Lat  \
count   17892.000000              8190          17892  17892.000000
unique           NaN               128            163           NaN
top              NaN  Diamond Princess             US           NaN
freq             NaN               126           3654           NaN
mean    13191.500000               NaN            NaN     26.287693
std      7624.675152               NaN            NaN     22.935092
min         1.000000               NaN            NaN    -41.454500
25%      6596.250000               NaN            NaN     13.145425
50%     13191.500000               NaN            NaN     32.985550
75%     19786.750000               NaN            NaN     42.501575
max     26382.000000               NaN            NaN     71.706900

                Long        Date  ConfirmedCases    Fatalities  \
count   17892.000000       17892    17892.000000  17892.000000
unique           NaN          63             NaN           NaN
top              NaN  2020-01-22             NaN           NaN
freq             NaN         284             NaN           NaN
mean        4.766191         NaN      325.207523     11.974737
std        79.923261         NaN     3538.599684    174.346267
min      -157.498300         NaN        0.000000      0.000000
25%       -71.516375         NaN        0.000000      0.000000
50%         9.775000         NaN        0.000000      0.000000
75%        64.688975         NaN       10.000000      0.000000
max       174.886000         NaN    69176.000000   6820.000000

        day_from_jan_first          temp           min           max  \
count          17892.00000  17892.000000  17755.000000  17876.000000
unique                 NaN           NaN           NaN           NaN
top                    NaN           NaN           NaN           NaN
freq                   NaN           NaN           NaN           NaN
mean               53.00000     54.849313     45.630262     64.380191
std                18.18475     22.306125     22.900739     22.310919
min                22.00000    -27.200000    -45.400000    -23.800000
25%                37.00000     38.800000     30.200000     47.500000
50%                53.00000     53.900000     44.400000     64.800000
75%                69.00000     76.800000     67.500000     84.600000
max                84.00000     97.300000     88.200000    109.600000

                stp           slp          dewp            rh            ah
```

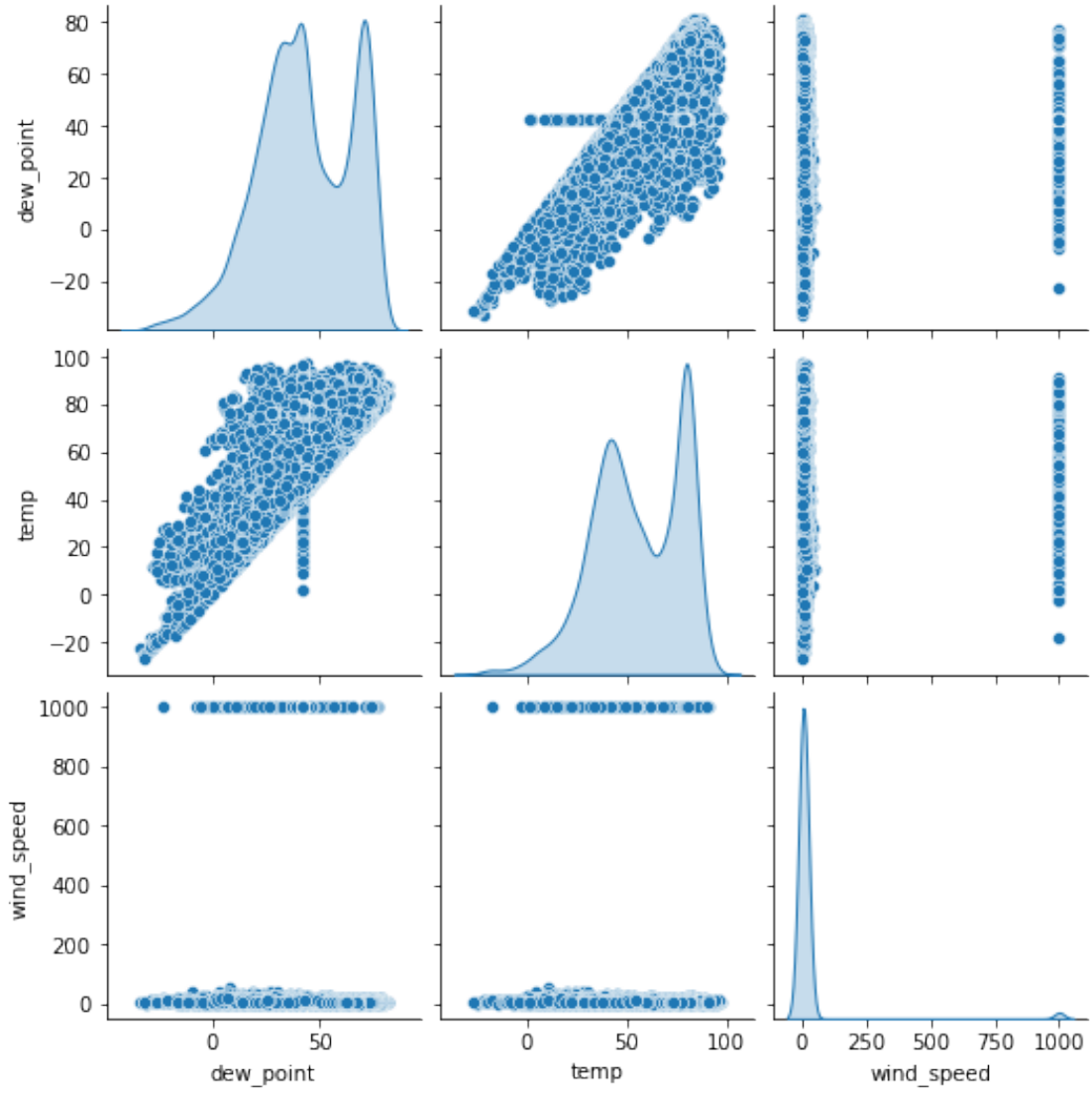
```
\
count   17892.000000  10945.000000  17274.000000  17274.000000  1.727400e+04
unique           NaN           NaN           NaN           NaN           NaN
top              NaN           NaN           NaN           NaN           NaN
freq             NaN           NaN           NaN           NaN           NaN
mean      702.306416   1016.581023     42.353630      0.665443           inf
std       428.769343      8.490953     22.399517      0.191092           NaN
min         0.000000    968.900000    -33.100000      0.053782 -2.374315e+01
25%        20.700000   1011.300000     27.000000      0.560904  1.161556e-01
50%       976.600000   1016.000000     40.800000      0.704800  1.932966e-01
75%       999.900000   1021.600000     63.500000      0.801220  2.329961e-01
max       999.900000   1051.700000     81.100000      1.000000           inf

                wdsp          prcp           fog
count   17892.000000  17892.000000  17892.000000
unique           NaN           NaN           NaN
top              NaN           NaN           NaN
freq             NaN           NaN           NaN
mean       25.521104      7.826334      0.336631
std       136.295573     26.740543      0.472571
min         0.000000      0.000000      0.000000
25%         3.500000      0.000000      0.000000
50%         5.600000      0.000000      0.000000
75%         8.700000      0.030000      1.000000
max       999.900000     99.990000      1.000000

df.dtypes

Id                     int64
Province/State        object
Country/Region        object
Lat                  float64
Long                 float64
Date                  object
ConfirmedCases       float64
Fatalities           float64
day_from_jan_first     int64
temp                 float64
min                  float64
max                  float64
stp                  float64
slp                  float64
dewp                 float64
rh                   float64
ah                   float64
wdsp                 float64
prcp                 float64
fog                    int64
dtype: object
```

```
new = pd.DataFrame()

new['dew_point']=df['dewp'].fillna(42.35)

new['dew_point'].isnull().sum()

0

new.head()

   dew_point
0       27.4
1       22.8
2       34.5
3       37.8
4       33.2

new['temp'] = df['temp']

new['wind_speed'] = df['wdsp']

new.head()

   dew_point  temp  wind_speed
0       27.4  42.6         9.4
1       22.8  42.0        14.9
2       34.5  40.1        10.4
3       37.8  46.0         6.1
4       33.2  42.8        10.8

mean_temp = new['temp'].mean()
median_temp = new['temp'].median()
print("mean temp:", mean_temp)
print("median temp:", median_temp)

mean temp: 54.849312541918245
median temp: 53.9

mean_windspeed = new['wind_speed'].mean()
median_windspeed = new['wind_speed'].median()
print("mean windspeed:", mean_windspeed)
print("median windspeed:", median_windspeed)

mean windspeed: 25.521104404203115
median windspeed: 5.6

df['Country/Region'].value_counts()

US            3654
China         2079
Canada         693
Australia      567
France         504
```

```
              ...
Greenland       63
Guadeloupe      63
Guam            63
Guatemala       63
Zambia          63
Name: Country/Region, Length: 163, dtype: int64
```

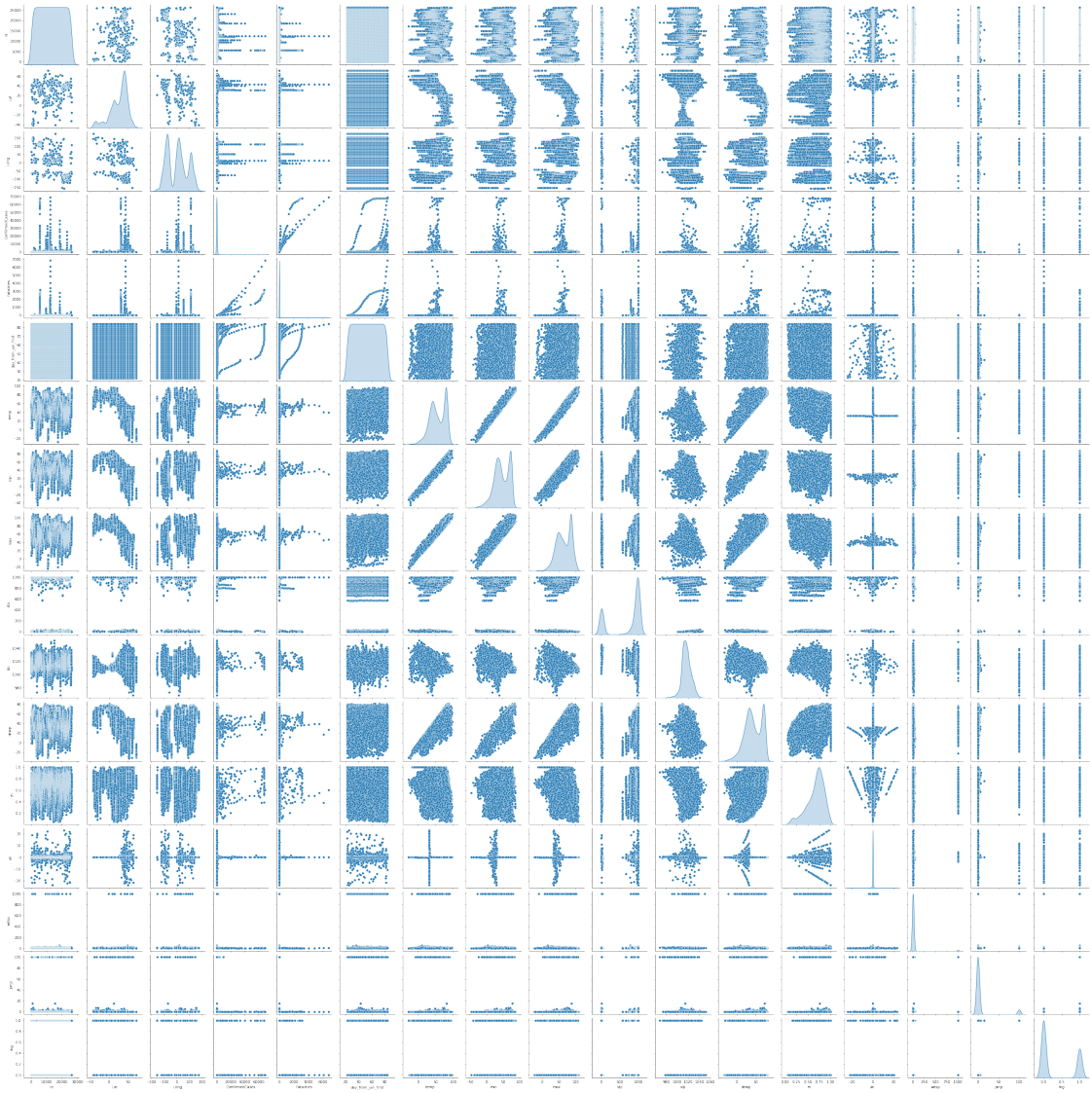
```
sns.pairplot(new,diag_kind='kde')
```

```
<seaborn.axisgrid.PairGrid at 0x27e0dde5430>
```



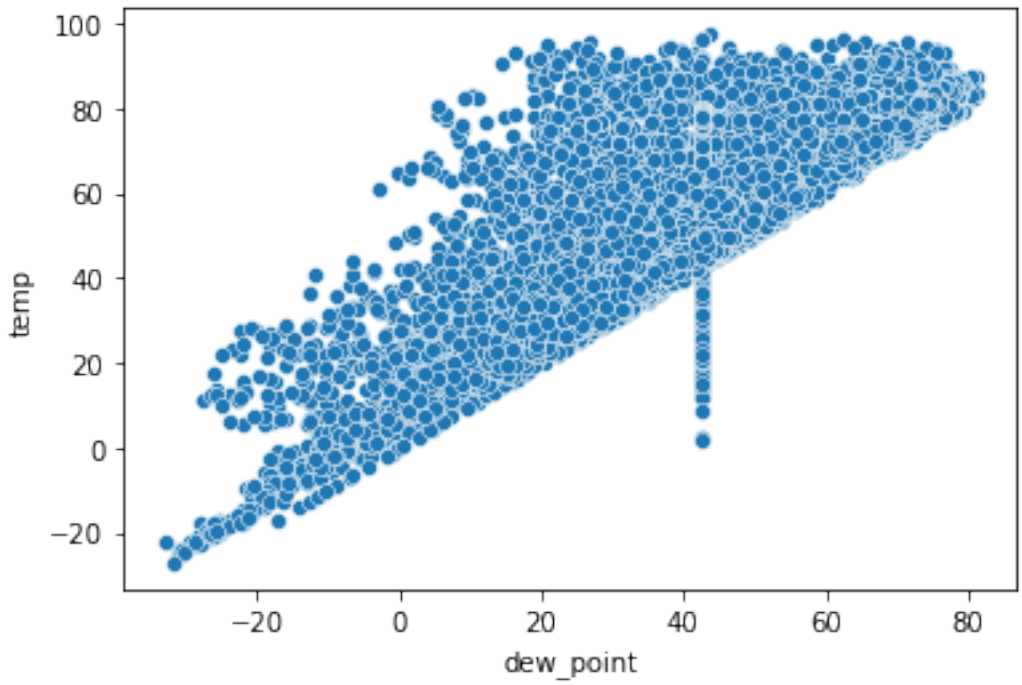

```
sns.pairplot(df,diag_kind='kde')
```

```
<seaborn.axisgrid.PairGrid at 0x27e0f3dc400>
```



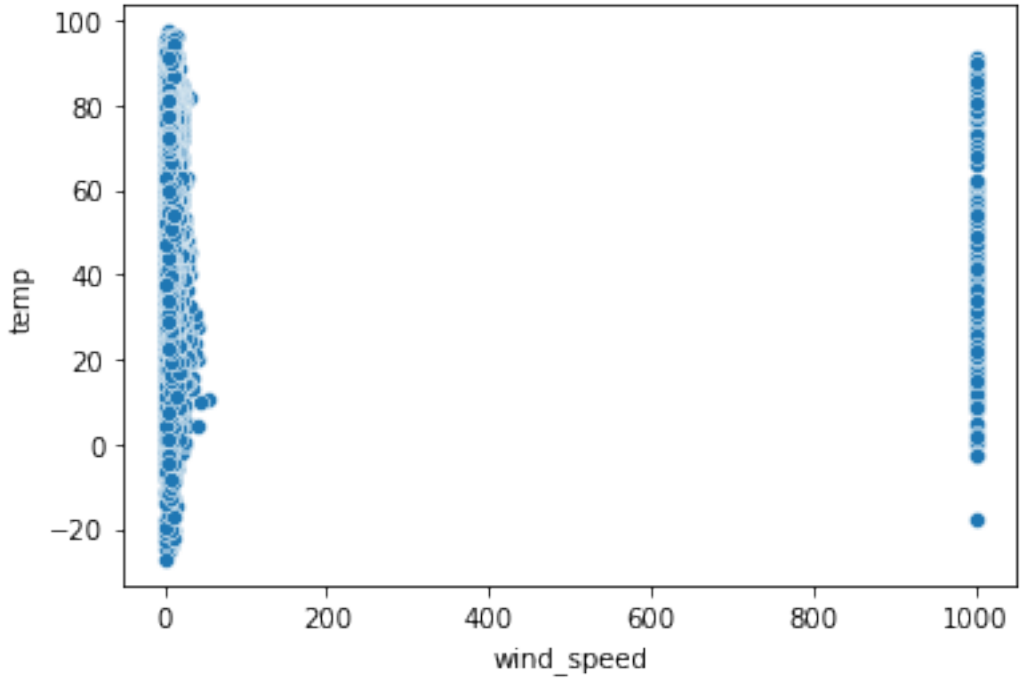
```
sns.scatterplot(x=new['dew_point'],y=new['temp'])
```

```
<AxesSubplot:xlabel='dew_point', ylabel='temp'>
```




```
sns.scatterplot(x=new['wind_speed'],y=new['temp'])
```

```
<AxesSubplot:xlabel='wind_speed', ylabel='temp'>
```




```
x = new[['dew_point','wind_speed']]
y = new['temp']

x_train,x_test,y_train,y_test = train_test_split(x,y,test_size = 0.25)
```

```
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(13419, 2)
(4473, 2)
(13419,)
(4473,)
```

```
model = LinearRegression()
```

```
model.fit(x_train,y_train)
```

```
LinearRegression()
```

```
y_pred = model.predict(x_test)
```

```
print("Model Score :",model.score(x_test,y_test)*100)
```

```
Model Score : 77.41954752915808
```