```
#!/usr/bin/env python
# coding: utf-8

# In[1]:


# import dependencies

import pandas as pd
import numpy as np


# In[2]:


data=pd.read_excel('Weather_forecast_file.xlsx')
data.head()


# In[3]:


pd.set_option('display.max_columns',None)


# In[4]:


data.head()


# In[5]:


data.shape


# In[6]:


for col, val in data.iteritems():
    print(col)
    print(val)


# ### Data Cleaning

# In[7]:


for col in data.columns:
    data[col]=data[col].str.strip("'")


# In[8]:


data.head()


# In[9]:


for col,val in data.iteritems():
    print(col," :",val.dtype)


# ### Task  : Remove Duplicates

# In[10]:


data.duplicated().any()


# In[11]:
```

```
np.where(data.duplicated())
```

```
# In[12]:
```

```
data.iloc[3900:3903]
```

```
# In[13]:
```

```
data.drop_duplicates(keep='first', inplace=True)
```

```
# In[14]:
```

```
data.duplicated().any()
```

```
# In[15]:
```

```
# Drop duplicate column
data.drop('Date1', axis=1, inplace=True)
```

```
# In[16]:
```

```
data.head()
```

```
# In[24]:
```

```
data['Dates']=pd.date_range(start='2022-01-01', periods=3902, freq='D')

# Created another date column with dates starting from 2022, this column will be used to store corrected years in the
# next step
```

```
# ### Task  : Correct year format (Data Cleaning)
```

```
# In[25]:
```

```
year=2021
for i in range( 0, len(data)):

    current=data[' Month'].iloc[i]
    prev=data[' Month'].iloc[i-1]
    if(current < prev):
        year=year + 1
    data['Dates'][i]=data['Date'].iloc[i].replace(data['Date'].iloc[i][0:4],str(year)) # storing corrected date
    print(year)
```

```
# In[26]:
```

```
data.head()
```

```
# In[27]:
```

```
pd.set_option('display.max_rows',None)
```

```
# ###  Typecasting Data to Suitable DataTypes
```

```
# In[28]:
```

```
data['Dates']=pd.to_datetime(data['Dates'],errors='coerce')
```

```
# In[29]:
```

```
for col  in data.iloc[:,1:-1]:
    data[col]=pd.to_numeric(data[col],errors='coerce')
```

```
# In[30]:
```

```
data.info()
```

```
# In[31]:
```

```
data.var()
```

```
# ###  Fixing Column Labels
```

```
# In[32]:
```

```
data.head()
```

```
# In[33]:
```

```
cols=list(data.columns)
cols[0],cols[-1]=cols[-1],cols[0]
cols
```

```
# In[34]:
```

```
data=data.reindex(columns=cols) #fixing column order
```

```
# In[35]:
```

```
col1=['Dates', 'Temperature', 'Average humidity (%',
       'Average dewpoint (Â°F', 'Average barometer (in',
       'Average windspeed (mph', 'Average gustspeed (mph',
       'Average direction (Â°deg', 'Rainfall for month (in',
       'Rainfall for year (in', 'Maximum temperature (Â°F',
       'Minimum temperature (Â°F', 'Maximum humidity (%',
       'Minimum humidity (%', 'Maximum pressure', 'Minimum pressure',
       'Maximum windspeed (mph', 'Maximum gust speed (mph',
       'Maximum heat index (Â°F', 'Month', 'Diff_pressure']
```

```
# In[36]:
```

```
data=data.rename(columns=str.strip) # removing white spaces from column names
```

```
# In[37]:
```

```
data.head()
```

```
# ### Removing redundant column
```

```
# In[38]:
```

```
data['Maximum rain per minute'].value_counts()


# In[39]:


data.drop(columns=['Date','Maximum rain per minute'],inplace=True)


# In[40]:


data.head()


# ### Check for Null Values

# In[41]:


data.isna().sum()


# In[42]:


print(np.where(data.Dates.isna()))
print(np.where(data['diff_pressure'].isna()))


# In[43]:


data.iloc[3750:3753]


# In[44]:


data.dropna(inplace=True)


# In[45]:


type(data)


# In[46]:


data.head()


# In[47]:


# data.to_csv('Weather_Forecasting.csv', encoding='utf-8')


# In[48]:


data.drop('Month', axis=1, inplace=True)


# In[49]:


data.head()


# ### Forecasting
# 1. Forecasting for 2034, 2035
# 2. Using FbProphet (no need to check stationarity & create differencing, Prophet takes care of everything)

# In[50]:
```

```
# pip install prophet


# In[51]:


# pip list


# In[52]:


# pip install --upgrade plotly


# In[53]:


from prophet import Prophet
from sklearn.metrics import mean_squared_error


# In[54]:


df=data[['Dates','Temperature']]
df.columns=['ds','y']
df.head()


# In[55]:


m=Prophet()
m.fit(df)
future=m.make_future_dataframe(periods=887, freq='D',include_history=False)
forecast=m.predict(future)


# In[56]:


forecast.head()


# In[57]:


df1=forecast[['ds','yhat']]
df1.columns=['Dates','Temperature']
df1.head()


# In[58]:


df=data[['Dates','Average humidity (%']]
df.columns=['ds','y']
df.head()


# In[59]:


m=Prophet()
m.fit(df)
future=m.make_future_dataframe(periods=887, freq='D', include_history=False)
forecast=m.predict(future)
forecast.head()


# In[60]:


df2=forecast[['ds','yhat']]
```

```
df2.columns=[['Dates','Average humidity (%']]
df2.head()


# In[61]:


data.head()


# In[62]:


df=data[['Dates','Average dewpoint (Â°F']]
df.columns=['ds','y']

m=Prophet()
m.fit(df)
future=m.make_future_dataframe(periods=887, freq='D', include_history=False)
forecast=m.predict(future)

df3=forecast[['ds','yhat']]
df3.columns=[['Dates','Average dewpoint (Â°F']]
df3.head()


# In[63]:


df=data[['Dates','Average barometer (in']]
df.columns=['ds','y']

m=Prophet()
m.fit(df)
future=m.make_future_dataframe(periods=887, freq='D', include_history=False)
forecast=m.predict(future)

df4=forecast[['ds','yhat']]
df4.columns=[['Dates','Average barometer (in']]
df4.head()


# In[64]:


data.columns


# In[65]:


df=data[['Dates','Average windspeed (mph']]
df.columns=['ds','y']

m=Prophet()
m.fit(df)
future=m.make_future_dataframe(periods=887, freq='D', include_history=False)
forecast=m.predict(future)

df5=forecast[['ds','yhat']]
df5.columns=[['Dates','Average windspeed (mph']]
df5.head()


# In[66]:


df=data[['Dates','Average gustspeed (mph']]
df.columns=['ds','y']

m=Prophet()
m.fit(df)
future=m.make_future_dataframe(periods=887, freq='D', include_history=False)
forecast=m.predict(future)

df6=forecast[['ds','yhat']]
df6.columns=[['Dates','Average gustspeed (mph']]
```

```
df6.head()


# In[67]:


df=data[['Dates','Average direction (Â°deg']]
df.columns=['ds','y']

m=Prophet()
m.fit(df)
future=m.make_future_dataframe(periods=887, freq='D', include_history=False)
forecast=m.predict(future)

df7=forecast[['ds','yhat']]
df7.columns=[['Dates','Average direction (Â°deg']]
df7.head()


# In[68]:


data.columns


# In[69]:


df=data[['Dates','Rainfall for month (in']]
df.columns=['ds','y']

m=Prophet()
m.fit(df)
future=m.make_future_dataframe(periods=887, freq='D', include_history=False)
forecast=m.predict(future)

df8=forecast[['ds','yhat']]
df8.columns=[['Dates','Rainfall for month (in']]
df8.head()


# In[70]:


df=data[['Dates','Rainfall for year (in']]
df.columns=['ds','y']

m=Prophet()
m.fit(df)
future=m.make_future_dataframe(periods=887, freq='D', include_history=False)
forecast=m.predict(future)

df9=forecast[['ds','yhat']]
df9.columns=[['Dates','Rainfall for year (in']]
df9.head()


# In[71]:


data.tail()


# In[72]:


df=data[['Dates','Maximum temperature (Â°F']]
df.columns=['ds','y']

m=Prophet()
m.fit(df)
future=m.make_future_dataframe(periods=887, freq='D', include_history=False)
forecast=m.predict(future)

df10=forecast[['ds','yhat']]
df10.columns=[['Dates','Maximum temperature (Â°F']]
df10.head()
```

```
# In[73]:


df=data[['Dates','Minimum temperature (Â°F']]
df.columns=['ds','y']

m=Prophet()
m.fit(df)
future=m.make_future_dataframe(periods=887, freq='D', include_history=False)
forecast=m.predict(future)

df10x=forecast[['ds','yhat']]
df10x.columns=[['Dates','Minimum temperature (Â°F']]
df10x.head()


# In[74]:


df=data[['Dates','Maximum humidity (%']]
df.columns=['ds','y']

m=Prophet()
m.fit(df)
future=m.make_future_dataframe(periods=887, freq='D', include_history=False)
forecast=m.predict(future)

df11=forecast[['ds','yhat']]
df11.columns=[['Dates','Maximum humidity (%']]
df11.head()


# In[75]:


df=data[['Dates','Minimum humidity (%']]
df.columns=['ds','y']

m=Prophet()
m.fit(df)
future=m.make_future_dataframe(periods=887, freq='D', include_history=False)
forecast=m.predict(future)

df12=forecast[['ds','yhat']]
df12.columns=[['Dates','Minimum humidity (%']]
df12.head()


# In[76]:


df=data[['Dates','Maximum pressure']]
df.columns=['ds','y']

m=Prophet()
m.fit(df)
future=m.make_future_dataframe(periods=887, freq='D', include_history=False)
forecast=m.predict(future)

df13=forecast[['ds','yhat']]
df13.columns=[['Dates','Maximum pressure']]
df13.head()


# In[77]:


df=data[['Dates','Minimum pressure']]
df.columns=['ds','y']

m=Prophet()
m.fit(df)
future=m.make_future_dataframe(periods=887, freq='D', include_history=False)
forecast=m.predict(future)
```

```
df14=forecast[['ds','yhat']]
df14.columns=[['Dates','Minimum pressure']]
df14.head()
```

```
# In[78]:
```

```
data.tail()
```

```
# In[79]:
```

```
df=data[['Dates','Maximum windspeed (mph']]
df.columns=['ds','y']

m=Prophet()
m.fit(df)
future=m.make_future_dataframe(periods=887, freq='D', include_history=False)
forecast=m.predict(future)

df15=forecast[['ds','yhat']]
df15.columns=[['Dates','Maximum windspeed (mph']]
df15.head()
```

```
# In[80]:
```

```
df=data[['Dates','Maximum gust speed (mph']]
df.columns=['ds','y']

m=Prophet()
m.fit(df)
future=m.make_future_dataframe(periods=887, freq='D', include_history=False)
forecast=m.predict(future)

df16=forecast[['ds','yhat']]
df16.columns=[['Dates','Maximum gust speed (mph']]
df16.head()
```

```
# In[81]:
```

```
df=data[['Dates','Maximum heat index (Â°F']]
df.columns=['ds','y']

m=Prophet()
m.fit(df)
future=m.make_future_dataframe(periods=887, freq='D', include_history=False)
forecast=m.predict(future)

df17=forecast[['ds','yhat']]
df17.columns=[['Dates','Maximum heat index (Â°F']]
df17.head()
```

```
# In[82]:
```

```
df=data[['Dates','diff_pressure']]
df.columns=['ds','y']

m=Prophet()
m.fit(df)
future=m.make_future_dataframe(periods=887, freq='D', include_history=False)
forecast=m.predict(future)

df18=forecast[['ds','yhat']]
df18.columns=[['Dates','diff_pressure']]
df18.head()
```

```
# In[88]:
```

```
df_list=[df1,df2,df3,df4,df5,df6,df7,df8,df9,df10,df10x,df11,df12,df13,df14,df15,df16,df17,df18]

# In[89]:

new_data=pd.concat(df_list, axis=1)

# In[90]:

new_data.head()

# In[93]:

new_data.iloc[:,[2,4,6,36]].head()

# In[94]:

new_data.drop(new_data.columns[[2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36]], axis=1,inplace=True)

# In[97]:

new_data.columns

# In[100]:

new_data.columns=  ['Dates', 'Temperature', 'Average humidity (%', 'Average dewpoint (Â°F',
       'Average barometer (in', 'Average windspeed (mph',
       'Average gustspeed (mph', 'Average direction (Â°deg',
       'Rainfall for month (in', 'Rainfall for year (in',
       'Maximum temperature (Â°F', 'Minimum temperature (Â°F',
       'Maximum humidity (%', 'Minimum humidity (%', 'Maximum pressure',
       'Minimum pressure', 'Maximum windspeed (mph', 'Maximum gust speed (mph',
       'Maximum heat index (Â°F', 'diff_pressure']

# In[101]:

new_data.head()

# In[102]:

# new_data.to_csv("Forecasted_data.csv", encoding='utf-8')

# In[109]:

# new_data.to_excel("Forecasted_data_xl.xlsx", encoding='utf-8')

# In[106]:

weather_complete=pd.concat([data, new_data], ignore_index=True) # Full Data: Historical + Forecasted
weather_complete.tail()

# In[107]:

# weather_complete.to_csv("Weather_Complete.csv", encoding='utf-8')
```