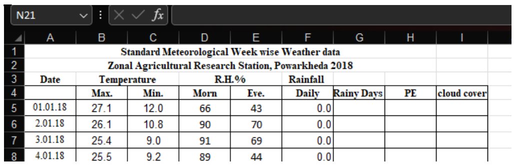
Processing of Collected Weather data (1996 – 2020)

Imported Libraries & Modules :

1. Matplotlib.pyplot
2. Seaborn
3. Pandas
4. Numpy
5. OS module

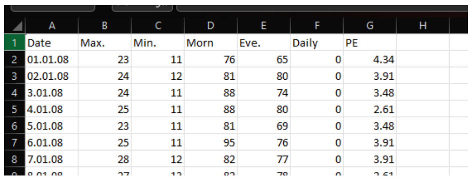
STEP 1 : Data Alignment

To arrange the data in a manner that pandas can use it to create a DataFrame & combined csv file that include weather data of all years (1996 to 2020) without any error I have used Mircosoft Excel sheet (drag , drop , cut , paste & basic things)



	A	B	C	D	E	F	G	H	I
1	Standard Meteorological Week wise Weather data								
2	Zonal Agricultural Research Station, Powarkheda 2018								
3	Date	Temperature		R.H.%		Rainfall			
4		Max.	Min.	Morn	Eve.	Daily	Rainy Days	PE	cloud cover
5	01.01.18	27.1	12.0	66	43	0.0			
6	2.01.18	26.1	10.8	90	70	0.0			
7	3.01.18	25.4	9.0	91	69	0.0			
8	4.01.18	25.5	9.2	89	44	0.0			

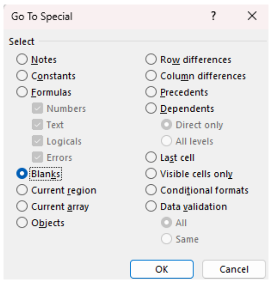
Un-Structured Arrangement



	A	B	C	D	E	F	G	H
1	Date	Max.	Min.	Morn	Eve.	Daily	PE	
2	01.01.08	23	11	76	65	0	4.34	
3	02.01.08	24	12	81	80	0	3.91	
4	3.01.08	24	11	88	74	0	3.48	
5	4.01.08	25	11	88	80	0	2.61	
6	5.01.08	23	11	81	69	0	3.48	
7	6.01.08	25	11	95	76	0	3.91	
8	7.01.08	28	12	82	77	0	3.91	

Structured Arrangement

Used Function's "Go to Special" in Excel Sheet to remove the blank row's between the data :



Go To Special

Select: [✓ Blanks]

Press, Ctrl + G > Special > Blank > ## Blank Row will be highlited ## > ## Select & Delete whole row ##
This Method will deleted all the blank row from sheet.

After Arranging the data in a Structured form , I renamed the column that have similar column (ET – Evapotranspiration & PE – Potential Evapotranspiration) since both are related to evaporation I renamed all column named ET to PE.
After this I exported each tab form xls to csv format & I got aligned data for each year in CSV file format.

STEP 2 : Creating a single csv file & removing unwanted rows & columns :

- Use for loop from python to store each year in a single list called "all_data".
- To skip year 2006 & 2007 use "continue" under loop (I have skiped this two year because this contain average data of 7 days represented as week, i.e 52 week for 1 year thus 52 rows for a csv file) .
- All csv's are stored in a single folder called "date wise csv" , join the address of this filder with the name of the each csv file with the help of join function form OS module .
- Remove the "Unnamed" columns, this column get created when extra index columns are saved in csv files unintensionally .
- Remove Row's which have "Average" in "Date" Column since they are not needed & are average of 7 days.
- Finally append the list to save.

STEP 3 : Creating DataFrame :

- Combine all the dataFrame from the list into a single dataframe using concat().
- Save the combined dataFrame into a csv file called "combined_weather_data.csv".
- Check weather the data is stored correctly using head() & tail() function.
- Load the combined_weather_data.csv into a variable called "weather".

STEP 4 : Cleaning of data

- “Rainy Days” column can only contain 0 & 1 , but it have values NaN , 20 , 11, which must be removed.
- Use isin() & isna() function to select & ~ to remove the selected values .
- Remove Rows with values "Average" in column "Date“.
- Remove rows with blank spaces in column "Date“.
- Removing Duplicate row using “data” column.
- Check for any unwanted values in “Rainy Days” column since its label it must contain 0 & 1 , remove others if found .
- Remove the column “cloud cover” it mostly contain NaN values (approx. 7300 values) .

STEP 5 : Convert into numeric values

- Conver all the data into numeric values since it’s type object we cannot do math in the data.
- weather['Max.'] = pd.to_numeric(weather['Max.'], errors='coerce’).
- Coerce here means it will convert into Null values if not convertible into numeric values.
- Do same function for all columns .

STEP 6 : Fill Median Values in place of Null Values

- Calculate Median for each table.
- Like this : median_Max = weather['Max.'].median()
- & then replace Null values with the respective Median value of that column.
- Like this : weather['Max.'] = weather['Max.'].fillna(median_Max)

STEP 7 : Check for Null values (If Any) in the weather data

- Use isnull() & sum() function to see the the column wise Null values .

STEP 8 : Export the clean data into a csv file

- To export the data into csv format use .to_csv() function.
- weather.to_csv('weather_clean_dataSet.csv', index=False)

STEP 9 : Split Train & Test data

- Remove columns Year & Date since they are not features of the data.
- Choose a large portion of dataset to train model & small portion to test the model.