Top of Form

Bottom of Form

|  |  |
| --- | --- |
|  | OUTPUT ENERGY OF WINDMILL USING WEATHER CONDITION  **LIBRARY FILES USED:**  import numpy as np |
|  | import pandas as pd |
|  | import csv |
|  | import os |
|  | import matplotlib.pyplot as py |
|  | import statistics  **NUMPY:**  NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.  **PANDAS:**  Pandas is an open-source library that is built on top of NumPy library. It is a Python package that offers various data structures and operations for manipulating numerical data and time series. It is mainly popular for importing and analyzing data much easier.  **CSV:**  The use of this lib is to verify and get datas from files with extension (.csv ).  **MATPLOTLIB.PYPLOT:**  This lib helps in plotting graph.  **STATISTICS:**  Python's statistics is a built-in Python library for descriptive statistics. You can use it if your datasets are not too large or if you can't rely on importing other libraries.  **FUNCTIONS:**  In this program I have used to functions one def plotfun and the other is def findmean the use  Of the functions in my program is :  **def plotfun** -> to get data to plot graph.  **def findmean** -> used to get data for calculating mean. |
|  | def plotfun(Speed,Powercurve,start,end): |
|  | WindSpeed=[] |
|  | for i in range(start,end+1): |
|  | WindSpeed.append(Speed[i]) |
|  |  |
|  | curve=[] |
|  | for i in range(start,end+1): |
|  | curve.append(Powercurve[i]) |
|  | py.plot(WindSpeed,curve)  the speed of the wind appended inside a list call windspeed. The theoratical power curve is appended inside the list called curve.  I used py.plot(x,y)  x-windspeed  y-curve  py.plot(windspeed,curve) is typed to plot the graph. |
|  |  |
|  | def findmean(arr,start,end): |
|  | l=[] |
|  | for i in range(start,end): |
|  | l.append(arr[i]) |
|  | x = statistics.mean(l) |
|  | print("the energy output of the windmill is ",x) |
|  | The theoratical power curve is appended inside the list called l. and mean is found |
|  |  |
|  | 1 for dirname, \_, filenames in os.walk('/kaggle/input'): |
|  | 2 for filename in filenames: |
|  | 3 print(os.path.join(dirname, filename)) |
|  | 4 data=pd.read\_csv(r"C:\Users\aruns\Desktop\WINDTURBINE\data.csv.zip") |
|  | 6 new\_data=({ |
|  | 7 "Time":data['Date/Time'], |
|  | 8 "Power":data['LV ActivePower (kW)'], |
|  | 9 "Speed":data['Wind Speed (m/s)'], |
|  | 10 "Power\_curve":data['Theoretical\_Power\_Curve (KWh)'], |
|  | 11 "Direction":data['Wind Direction (°)'] |
|  | 12 })  The above is used to read the file. You can see in the 4th line the data file is taken bye pasting the location. And each data in it read using different names using those names we call the data inside the programm. |
|  |  |
|  | q=int(input("Do you need to calculate the output energy of windmill in hours or in minutes or in days?(1=hours/0=minutes/2=days): ")) |
|  | if q==1: |
|  | a=int(input("Number of hours prediction you need: ")) |
|  | b=6#i have a constant value this is the number of columns because the mini value i took is 1hour=6columns |
|  | c=a\*b |
|  | elif q==0: |
|  | a=int(input("Number of minutes prediction you need: ")) |
|  | b=10#though the given data has prediction for 10minutes so i took 10min=1column |
|  | c=a//b |
|  | #the bellow code will calculate in days |
|  | elif q==2: |
|  | a=int(input("Number of days prediction you need: ")) |
|  | b=144 |
|  | c=a\*b  in this I have used ifelse code in this code you can set the time until which day are until which hour we wnt the output.  **IFELSE:**  An if else statement in programming is a conditional statement that runs a different set of statements depending on whether an expression is true or false |
|  |  |
|  | plotfun(new\_data["Speed"],new\_data["Power\_curve"],0,c) |
|  | findmean(new\_data["Power\_curve"],0,c)  in this code the two functions are called. |