

kpis

May 16, 2023

1 KPIS :

Pour calculer les KPIs suivants : Les temps net et brute de fonctionnement, MTBF, MTTR et Disponibilite materiel.

```
[46]: import pandas as pd
import numpy as np

def kpi(file_path):

    # Charger le fichier CSV dans un DataFrame :
    df = pd.read_csv(file_path)

    #le temps de base heurs par jours:
    tmp_base = 6
    #les heurs de fonctionnements par jours :
    temp_fjours = 12

    #Calculer le temps de maintenance predictive :
    temps_main_predict = [np.random.uniform(10,30) for i in range(len(df))] * \
↳df['failure']

    #calculer le temps de maintenances corrective :
    temps_main_corr = [np.random.uniform(15, 50) for i in range(len(df))] * \
↳df['realFail']

    #calculer le temps net de fonctionnement :

    List_jours2 =[31,29,31,30,31,30,31,31,30,31,30,31]
    for elem in List_jours2:
        temps_net = tmp_base*elem + (temps_main_predict/60 *elem)

    #calculer le temps brut de fonctionnement :

    diff = df['failure']-df['realFail']
    rend = [i for i in range(len(diff))]
    rend = np.array(rend)
```

```

temps_brut = [i for i in range(len(diff))]
temps_brut = np.array(temps_brut)
for i in range(len(diff)):

    if (diff[i]==0) :
        for elem in List_jours2:
            temps_brut[i]=temps_net[i] + (temps_main_corr[i]/60 *elem)
            rend[i]= (temps_net[i]/temps_brut[i] )*100
    elif (diff[i]>0):
        for elem in List_jours2:
            temps_brut[i]=temps_net[i] + (temps_main_predict[i]/60
↳*elem)

            rend[i]= (temps_net[i]/temps_brut[i] )*100
    elif (diff[i]<0):
        #calculer le temps brut :
        for elem in List_jours2:
            temps_brut[i] = temps_net[i] + (temps_main_corr[i]/60 *elem)
            rend[i]=(temps_net[i]/temps_brut[i])*100

#calcul des KPIs
#calculer MTBF: nombre heures de fonctionnement / nombre de pannes

# mtbf = temp_fjours*nombre_de_jours/df['realFail']
List_jours = [31,29,31,30,31,30,31,31,30,31,30,31]
for elements in List_jours:
    mtbf = temp_fjours*elements/df['realFail']

#calculer MTTR :

mttr = temps_main_corr/df['realFail']

#disponibilite materielle
dm = mtbf /(mtbf+mttr)

# Ajouter les colonnes des KPIs au DataFrame
df['temps_fonctionnement_parJ'] = temp_fjours
df['temps_utile/inutile'] = diff
df['temps_brut'] = temps_brut
df['temps_net'] = temps_net
df['productivite'] = rend

df['mtbf'] = mtbf
df['mttr'] = mttr
df['disponibilité_materielle'] = dm

```

```
# Enregistrer le nouveau fichier CSV avec les nouvelles colonnes
new_file_path = file_path.split('.csv')[0] + '_with_kpisfs33.csv'
df.to_csv(new_file_path, index=False)
```

```
# Retourner le DataFrame avec les nouvelles colonnes
return df
```

```
df_with_kpisf33 = kpi('results.csv')
df_with_kpisf33
```

```
[46]:
```

	month	failure	realFail	temps_fonctionnement_parJ	temps_utile/inutile	\
0	1	6	11	12	-5	
1	2	6	9	12	-3	
2	3	4	6	12	-2	
3	4	4	9	12	-5	
4	5	6	11	12	-5	
5	6	5	9	12	-4	
6	7	5	7	12	-2	
7	8	12	8	12	4	
8	9	16	15	12	1	
9	10	11	4	12	7	
10	11	4	6	12	-2	
11	12	2	8	12	-6	

	temps_brut	temps_net	productivite	mtbf	mttr	\
0	463	246.516557	53	33.818182	38.200871	
1	335	237.064654	70	41.333333	21.151067	
2	306	222.638251	72	62.000000	27.109176	
3	372	243.687728	65	41.333333	27.660030	
4	332	217.144356	65	33.818182	20.376207	
5	377	239.164918	63	41.333333	29.698185	
6	362	238.574237	65	53.142857	34.304749	
7	440	313.067423	71	46.500000	35.441313	
8	637	411.913047	64	24.800000	39.095139	
9	375	280.582260	74	93.000000	19.872229	
10	316	223.287447	70	62.000000	30.108720	
11	355	199.865785	56	46.500000	37.772304	

	disponibilité_materielle
0	0.469573
1	0.661498
2	0.695776
3	0.599091
4	0.624016
5	0.581901
6	0.607711
7	0.567479

8	0.388136
9	0.823940
10	0.673118
11	0.551783

2 Si on effectue pas la maintenance preductive..

```
[1]: import pandas as pd
import numpy as np

# Charger le fichier CSV dans un DataFrame :
df2= pd.read_csv('results.csv')

# Remplacer la colonne failure avec 0:
df2['failure'] = 0

# Enregistrer les donnees dans un nouveau fichier CSV
df2.to_csv('results_with_any_feature.csv', index=False)
```