

## Istruzioni per l'uso:

Il progetto è stato generato e compilato su <https://colab.research.google.com/>

Se si volesse solamente vedere il codice e le print del progetto Flight, importare il file .ipynb su Colab, altrimenti:

1) Installare le librerie mancanti con questa parte di codice(Jupyter o Colab):`!pip install "nome_libreria"`

1A) Aggiornare le librerie con questa parte di codice(Jupyter o Colab): `!pip install "nome_libreria" --upgrade`

2)La libreria "montecarlo" sarà installata con: `!pip install pandas_montecarlo`

3)La libreria "montecarlo" sarà aggiornata con: `!pip install pandas_montecarlo --upgrade --no-cache-dir` o  
`!pip install pandas_montecarlo --upgrade`

4)Importare il dataset da **Jupyter Notebook** in questo modo:

`-URL ="flights.csv"`

4B) Importare il dataset da **Colab** in questo modo:

```
-URL ='/content/drive/My Drive/flights.csv'
```

\*\*\*\*\*N.B Uno esclude l'altro, ovvero se si è su Jupyter Notebook cancellare l'altra riga di codice (Colab) e viceversa\*\*\*\*\*

Esempio: Jupyter Notebook

```
URL = "flights.csv"
```

```
# URL = '/content/drive/My Drive/flights.csv'
```

Colab:

```
#URL = "flights.csv"
```

```
URL = '/content/drive/My Drive/flights.csv'
```

5) Se si utilizza Colab importare e caricare il Dataset "flights.csv" sul proprio Google Drive, prima dei punti **4B**

6) Il caricamento del Dataset avviene, su **Jupyter Notebook**, solo se il file Jupyter e il Dataset sono nella stessa cartella

7) Cancellare questa parte di codice (si trova all'inizio del codice) solo se si utilizza **Jupyter Notebook**:

```
#from google.colab import drive
```

```
#drive.mount('/content/drive')
```

8) Se invece si è su **Colab non** cancellare il punto 7

9)All'interno della funzione "Delay\_creator\_with\_MonteCarlo" in questa parte del codice:

mc = delay\_of\_ind\_3.montecarlo(sims=10) --> impostare il numero di sims(simulazioni) che si vuole utilizzare

CODICE: La parte interessata, ha il simbolo "\*"

```
-----  
def Delay_creator_with_MonteCarlo(df,column):  
    delay_serie = pd.Series()  
    for delay_of_ind in df[str(column)]:  
        delay_of_ind_2 = pd.Series(delay_of_ind) # trasforma riga in serie  
        delay_of_ind_3 = delay_of_ind_2.transpose() # serie in colonna  
        mc = delay_of_ind_3.montecarlo(sims=10) ***** # .montecarlo è  
una libreria Python  
        stats = mc.stats  
        stats_std = stats['max'] #prendo il massimo dei ritardi  
        stats_std = pd.Series([stats_std]) #lo metto in una serie di lista per motivi computazionali  
        delay_serie = delay_serie.append(stats_std,ignore_index=True)  
        delay_df = pd.DataFrame(delay_serie)  
        delay_df = delay_df.rename(columns={0: "Delay_Series"})  
        df1 = pd.concat([df, delay_df], axis=1)  
    return df1  
-----
```

10)Genetic Algorithm --> In questa parte del codice(che si trova nella funzione), impostare il numero di iterazioni che si vuole utilizzare:

for i in range(100): --> impostare il numero di iterazioni che si vuole utilizzare

selection= select\_ind\_probabilities(list\_path, "Probability", "weighted", 10) --> impostare o "weighted", "besthalf" o "random" se si vuole utilizzare una probabilità, cambiando il metodo di selezione per il Genetico

CODICE: La parte interessata, ha il simbolo "\*"

```
-----  
  
#iterate fitness function  
elite = pd.DataFrame()  
df_generale = list_path  
F,Q = [],[] #liste che servono per le visualizzazioni  
lista_valori_max, lista_indici_max = [],[]  
  
for i in range(500): *****  
  
    selection= select_ind_probabilities(list_path, "Probability", "weighted", 10)  
    *****  
  
    selection_path = selection.Path  
    selection_path = selection_path.to_frame()  
    selection_path.rename(columns={0:'Path'}, inplace=True)  
  
    ...
```

-----