

Monte Carlo Algorithm

“Analysis Open Ticket in Programmer at Indotaichen Textile  
Using Monte Carlo”

Oleh:

Tobias Mikha Sulistiyo

### 1. Apa itu Monte carlo?

Monte carlo merupakan algoritma yang menggunakan metode probabilitas atau random untuk menyelesaikan masalah prediksi. Metode simulasi Monte carlo menghasilkan berbagai hasil yang mungkin dengan probabilitas setiap hasil yang terjadi. Metode ini digunakan dalam bidang fisika dan riset.

### 2. Cara kerja Monte carlo

- Mendefinisikan apa yang ingin diprediksi dari suatu data
- Memberikan batasan/data cleansing pada data tersebut
- Menghitung peluang munculnya dari suatu data
- Menghitung estimasi data yang muncul.
- Untuk meningkatkan akurasi, data masukan bisa diperbanyak (Perbanyak sample data)

### 3. Code & Algorithm

Note: Untuk Simulasi kali ini, saya menggunakan python untuk menjalankan algoritma ini

#### 1. Import library

```
import pandas as pd
import numpy as np
from google.colab import drive
```

#### 2. Mount data, untuk data ini saya letakkan di google drive, sehingga saya perlu import library drive.

```
drive.mount('MyDrive/')
```

#### 3. Melihat nilai data yang sudah diimport

```
data.describe()
```

|        | IDENTIFIEDDATE             | PMBOMCODE       | TGL_BUAT_BD                | NO_BD         | NO_WO       | NO_PPA      | DEPT | DEPT22 | JENI |
|--------|----------------------------|-----------------|----------------------------|---------------|-------------|-------------|------|--------|------|
| count  | 370                        | 370             | 370                        | 370           | 370         | 370         | 370  | 370    |      |
| unique | 366                        | 105             | 366                        | 370           | 370         | 370         | 23   | 16     |      |
| top    | 2024-08-22-02.50.51.000000 | DITONLINE000059 | 2024-08-22-02.50.51.000000 | BDIT240002425 | BWO24007551 | PPA24097551 | QCF  | DIT    |      |
| freq   | 4                          | 87              | 4                          | 1             | 1           | 1           | 97   | 320    |      |

4 rows x 36 columns

#### 4. Dari nilai data tersebut, saya ingin melakukan prediksi untuk kolom “jenis permohonan” yang akan muncul

```
prob_permohonan = data['JENIS_PERMOHONAN'].value_counts(normalize=True)
print(round(prob_permohonan,2))
```

```
JENIS_PERMOHONAN
Penambahan    0.58
Perubahan     0.28
Perbaikan     0.08
Baru          0.06
Name: proportion, dtype: float64
```

Pada permintaan program, biasanya tiap permohonan memiliki jangka waktu yang berbeda, sehingga apabila sudah terdapat prediksi untuk jenis permohonan yang akan muncul di bulan/periode berikutnya, menjadi lebih mudah untuk menghitung estimasi waktu yang akan dikerjakan bagi programmer.

5. Selain jenis permohonan, departement yang mengajukan juga menjadi parameter

```
prob_dept = data['DEPT'].value_counts(normalize=True).sort_values(ascending=False)
print(round(prob_dept,2))
```

```
DEPT
QCF    0.26
DYE    0.14
DIT    0.08
MKT    0.06
GKG    0.06
LAB    0.05
PPC    0.04
GKJ    0.04
HRD    0.04
GAS    0.04
KNT    0.04
FIN    0.03
BRS    0.02
EXIM   0.02
TAS    0.01
GDB    0.01
YND    0.01
PCS    0.01
MNF    0.01
MTC    0.01
QAI    0.01
MRP    0.00
ACC    0.00
Name: proportion, dtype: float64
```

Departement yang mengajukan juga menjadi parameter untuk open ticket, karena tiap departement memiliki web masing masing sehingga dapat menjadi parameter apakah web tersebut sudah stabil atau belum

6. Disini saya membuat parameter inputan, supaya bisa saya kembangkan untuk menjadi website, untuk membantu menganalisa open ticket kedepannya. Saya handling juga supaya user hanya dapat memasukkan angka, dan angka yang diinputkan merupakan target open ticket yang akan dibuka untuk bulan berikutnya

```
while True:
    try:
        n = int(input('Masukkan Jumlah open ticket: '))
        break
    except ValueError:
        print("Input tidak valid. Harap masukkan angka yang valid.")
```

```
Masukkan Jumlah open ticket: 64
```

7. Untuk prediksi pertama adalah prediksi departemen yang memiliki kemungkinan untuk open ticket / request program

```
# Simulasi prediksi departemen
prediksi_dept = np.random.choice(prob_dept.index, size=n, p=prob_dept.values)
# Memasukkan hasil simulasi ke dalam DataFrame
simulasi_dept = pd.DataFrame({
    'DEPT': prediksi_dept
})
print(simulasi_dept['DEPT'].value_counts().head(6))
```

| DEPT |    |
|------|----|
| QCF  | 15 |
| DIT  | 9  |
| DYE  | 8  |
| LAB  | 6  |
| PPC  | 4  |
| KNT  | 4  |

Name: count, dtype: int64

8. Kemudian, melakukan prediksi untuk jenis permohonan yang kemungkinan akan keluar

```
# Simulasi prediksi untuk jenis permohonan yang akan keluar
prediksi_permohonan = np.random.choice(prob_permohonan.index, size=n, p=prob_permohonan.values)

# Memasukkan hasil simulasi ke DF
hasil_permohonan = pd.DataFrame({
    'PERMOHONAN': prediksi_permohonan
})

# Menghitung prediksi kemungkinan keluar
print(hasil_permohonan['PERMOHONAN'].value_counts())
```

| PERMOHONAN |    |
|------------|----|
| Penambahan | 38 |
| Perubahan  | 21 |
| Perbaikan  | 3  |
| Baru       | 2  |

Name: count, dtype: int64

4. Kesimpulan Monte Carlo & Simulasi

Dari hasil simulasi didapatkan bahwa kemungkinan departemen yang akan request program terbanyak untuk kedepannya adalah QCF, DIT, DYE, LAB, PPC, KNT.

Dan untuk jenis permohonan yang memiliki kemungkinan terbanyak untuk muncul adalah permintaan untuk perubahan dan penambahan.

Dari hasil algoritma dan kinerja program, tentunya monte carlo sangat memudahkan untuk melakukan prediksi. Tingkat akurasi dari algoritma monte carlo sangat berpengaruh dengan data inputan yang ada. Apabila data inputan memiliki banyak data outlier, data tersebut dapat mempengaruhi akurasi dan hasil dari Analisa dan kinerja program.

5. Lampiran

Link google colab:

<https://colab.research.google.com/drive/1eiDsILfSY05pMhNByrBAa2VaFLlRqiKi?usp=sharing>