|  |  |  |
| --- | --- | --- |
| **Data Noise, Manipulasi Data, Normalisasi Data** | | |
| Praktikan | **Aslab** | |
| Nama: xxxx  Nim: xxxx | Annur Hangga Prihadi | 065001800028 |
| Faiz Kumara | 065001800003 |

**PRAKTIKUM 2**

**DATA SAINS DAN ANALITIK**

Topik pertemuan praktikum kedua adalah mengetahui cara menemukan data missing bersifat noise dan membersihkan data noise tersebut, manipulasi data berdasarkan grup, manipulasi data berdasarkan filter, normalisasi Zscore dan Minmax.

**Latihan 1**

1. **Memasang library yang dibutuhkan**

|  |
| --- |
| **import sys**  **import psycopg2**  **import pandas as pd** |

1. **Koneksikan ke database**

|  |
| --- |
| **conn = psycopg2.connect(host="localhost", port = XXXX, database="dsda", user="postgres", password="XXXX")**  **conn.set\_session(autocommit=True)**  **cur = conn.cursor()**  **sql = "SELECT \* FROM public.NAMA\_TABEL"**  **cur.execute(sql)**  **baris = cur.fetchall()** |

1. **Listing raw dataset, rename, dan menampilkan n data**

|  |
| --- |
| order = pd.DataFrame([[ij for ij in i]for i in baris])  order.rename(columns={0:'order\_id',1:'customer\_id',2:'campaign\_id',3:'order\_date',4:'city',5:'state',6:'zipcode',7:'payment\_type',8:'total\_price',9:'num\_order',10:'num\_units'},inplace=True)  order.head(4) |

1. **Cek missing value/null**

|  |
| --- |
| **print(order.isnull().sum())** |

1. **Fix missing value/null dan cek missing value/null setelah dibersihkan**

|  |
| --- |
| **cleandata = order.dropna()**  **print(cleandata.isnull().sum())** |

1. **Cek dimensi raw data**

|  |
| --- |
| **order.shape** |

1. **Cek dimensi data yang sudah bersih**

|  |
| --- |
| **cleandata.shape** |

1. **Tampilkan n data yang sudah bersih**

|  |
| --- |
| **cleandata.head(4)** |

1. **Manipulasi berdasarkan grup data**

|  |
| --- |
| **data = order[['order\_id','customer\_id','campaign\_id','total\_price']]**  **idOrder = data.groupby('campaign\_id')['customer\_id'].nunique()**  **print('Jumlah Customer:\n',idOrder)** |

1. **Manipulasi berdasarkan filter data NY**

|  |
| --- |
| **man2 = cleandata[['order\_id','customer\_id','campaign\_id','payment\_type']]**  **stateNY = man2['campaign\_id'].loc[cleandata['state']=="NY"]**  **print(stateNY.value\_counts())** |

1. **Manipulasi berdasarkan filter data PA**

|  |
| --- |
| **statePA = man2['campaign\_id'].loc[cleandata['state']=="PA"]**  **print(statePA.value\_counts())** |

1. **Normalisasi Zscore (Sebelum)**

|  |
| --- |
| **zdata = cleandata.loc[:,['total\_price']]**  **zdata['total\_price'] = zdata['total\_price'].fillna(cleandata.groupby('campaign\_id')['total\_price'].transform('mean'))**  **print('Data sebelum normalisasi ZScore:\n',zdata)** |

1. **Normalisasi Zscore (Sesudah)**

|  |
| --- |
| **avgdata = zdata.mean()**  **stddata = zdata.std()**  **zdata = (zdata-avgdata)/stddata**  **print('Data setelah normalisasi ZScore:\n',zdata)** |

1. **Normalisasi Minmax (Sebelum)**

|  |
| --- |
| **mdata = cleandata.loc[:,['total\_price']]**  **mdata['total\_price'] = mdata['total\_price'].fillna(cleandata.groupby('campaign\_id')['total\_price'].transform('mean'))**  **print('Data sebelum normalisasi minmax:\n',mdata)** |

1. **Normalisasi Minmax (Sesudah)**

|  |
| --- |
| **min = 0**  **max = 1**  **mindata = mdata.min()**  **maxdata = mdata.max()**  **mdata = ((mdata-mindata)\*(max-min)/(maxdata-mindata))+min**  **print('Data setelah normalisasi minmax:\n',mdata)** |

**Latihan 2**

1. Manipulasikan data berdasarkan grup data dari kolom payment\_type untuk customer\_id!
2. Manipulasikan data berdasarkan grup data dari kolom city untuk customer\_id!
3. Manipulasikan data berdasarkan filter data dari kolom num\_units untuk kolom state yang dimana berisi value TX!
4. Manipulasikan data berdasarkan filter data dari kolom num\_order untuk kolom state yang dimana berisi value FL!

**Lampiran Screenshot hasil 1, 2, 3, dan 4**

|  |
| --- |
| **Input screenshot disini** |

**Makna dari masing-masing hasil di atas!**

|  |
| --- |
| **Ketik makna disini** |