Normalisasi Data Asuransi Kesehatan Menggunakan Decorator

Arafi Ramadhan Maulana (122450002)[1], Dwi Ratna Anggraeni (122450008)[2], Raid Muhammad Naufal (122450027)[3], Rayan Koemi Karuby (122450038)[4], Muhammad Deriansyah Okutra (122450101)[5]

Program Studi Sains Data, Fakultas Sains, Institut Teknologi Sumatera

Email :
arafi.122450002@student.itera.ac.id [1] , *dwi.122450008@student.itera.ac.id* [2] , *raid.122450027@student.itera.ac.id* [3] , *rayan.122450038@student.itera.ac.id* [4] , *mderiansyah.122450101@student.itera.ac.id* [5]

1. Pendahuluan

Jurnal ini membahas tentang proses normalisasi data menggunakan dataset asuransi dan analisisnya. Normalisasi data merupakan suatu pendekatan sistematis untuk meminimalkan redundansi data pada suatu database agar database tersebut dapat bekerja dengan optimal. Tujuan normalisasi database adalah untuk menghilangkan dan mengurangi redudansi data dan tujuan yang kedua adalah memastikan dependensi data (Data berada pada tabel yang tepat) (Prayitno, 2018).

Pada kali ini dilakukan penerapan konsep closure dengan sintaks *sugar, decorator,* dan teknik *Memoization. Closure* adalah sebuah fungsi yang bisa disimpan dalam variabel. Dengan menerapkan konsep tersebut, kita bisa membuat fungsi di dalam fungsi, atau bahkan membuat fungsi yang mengembalikan fungsi. *Closure* adalah fungsi tanpa nama dan dimanfaatkan untuk membungkus suatu proses yang hanya dipakai sekali atau dipakai pada blok tertentu saja (Prayogo, 2022).

Sintaks *sugar, decorator, dan memoization*, ketiganya merupakan alat penting dalam yang memperluas kegunaan dari closure. Sintaks sugar merupakan sintaks yang dapat membantu para developer menulis kode dengan mudah. (Buzulan, 2020). *Decorator* merupakan sebuah teknik yang dilakukan dengan cara mengambil sebuah fungsi, menambahkan fungsionalitasnya dan mengembalikannya (return) (Veerisetty, 2019).

Memoization merupakan teknik yang dilakukan untuk mempercepat performa aplikasi. *Memoization* menggunakan cache untuk return nilai yang didapat dari proses *run-time* yang sekiranya lama dan menggunakan sumber daya yang besar (Kahari, 2022).

Dalam konteks dataset asuransi yang digunakan, normalisasi data bertujuan untuk memastikan bahwa berbagai fitur atau variabel dalam dataset, seperti usia, biaya premi, dan jumlah klaim, memiliki skala yang serupa, sehingga memudahkan perbandingan dan analisis lebih lanjut.

Analisis yang akan dilakukan dalam jurnal ini diharapkan dapat memberikan pemahaman yang lebih baik tentang distribusi dan pola data dalam dataset asuransi, setelah melalui proses normalisasi.

2. Metode

2.1. *Pandas*

Pandas merupakan sebuah *library python* yang *open source* yang digunakan dalam analisis data. Struktur dasar pada pandas adalah data frame. *Series* dan *Data Frame* merupakan tipe struktur data dari pandas. Pandas menyediakan struktur data dengan cepat, fleksibel dan ekspresif yang dirancang untuk data relasional atau berlabel.

Berikut beberapa fungsi dari *pandas* yaitu menyelaraskan data sebelum dibandingkan ataupun penggabungan *dataset,* menangani data hilang, dan lain-lain.

2.1.1. *select_dtypes()*

Select_dtypes() merupakan salah satu fungsi yang terdapat pada pandas digunakan untuk mengembalikan *DataFrame* baru yang tidak memasukan atau tidak kolom-kolom dengan *data type* tertentu.

2.1.2. *update()*

Update() dalam pandas digunakan untuk merubah sebuah *data frame* dengan nilainya dari *data frame* lain, dimana hanya indeks dan label kolom yang cocok yang diperbarui.

2.1.3. *read_csv()*

Read_csv adalah salah satu fungsi pada *pandas* yang digunakan untuk membaca file csv sehingga kita dapat memanipulasi data dari file csv tersebut.

2.1.4. *describe()*

Describe di pandas digunakan untuk mendapatkan gambaran dari data kita dengan cepat dan umum. Describe dapat menampilkan rata-rata, standar deviasi, nilai minimum, nilai maksimum, kuartil 1, kuartil 2 atau median, kuartil 3, dan nilai maksimum.

2.2. *Wraps*

Wrapper merupakan penggunaan fungsi untuk mengubah perilaku fungsi atau yang sudah ada tanpa mengubah kode sumber asli fungsi tersebut. Wrapper sering digunakan untuk menambahkan fungsionalitas tambahan atau memodifikasi perilaku tanpa perlu mengubah kode yang sudah ada.

2.3. *normalize_data()*

Normalisasi merupakan teknik pra-pemrosesan yang berguna untuk mengubah data mentah menjadi data yang terstruktur dan meningkatkan efisiensi pembelajaran mesin. Ini berarti bahwa ketika data dikumpulkan dari berbagai sumber, data tersebut awalnya dalam bentuk mentah yang tidak dapat langsung digunakan untuk analisis atau pembelajaran mesin.

2.3.1. *decorated_function()*

Dalam *python,* dekorator merupakan pola desain yang memungkinkan kita dapat merubah fungsionalitas sebuah fungsi dengan membungkusnya dengan fungsi lain. Fungsi terluar disebut dekorator, yang mengambil fungsi asli sebagai argumen dan mengembalikan versi modifikasi dari fungsi tersebut.

2.4. *dataset()*

Dataset merupakan kumpulan sebagai kumpulan data yang disusun dalam format tertentu yang digunakan untuk melakukan analisis, pelatihan model, ataupun penelitian. Dataset menjadi fondasi utama yang digunakan dalam pengidentifikasian pola, pembangunan model, dan untuk memperoleh *insight* dari sebuah data.

2.5. statistika_deskriptif()

Statistika deskriptif adalah metode yang melibatkan pengumpulan dan penyajian data yang dapat memberikan informasi yang bermanfaat. Statistika deskriptif berfungsi untuk menggambarkan, meringkas dan mengumpulkan data agar lebih mudah digunakan dalam analisis data.

3. Pembahasan

Pada studi kasus ini, data yang digunakan merupakan dataset asuransi kesehatan yang memiliki tujuh variabel (*age, sex,* bmi, *children, smoker, region,* dan *charges).* Variabel *age*

mempresentasikan umur penerima asuransi dalam tahun, variabel *sex* mempresentasikan jenis kelamin penerima asuransi, variabel bmi mempresentasikan indeks massa tubuh, variabel *children* mempresentasikan jumlah anak yang ditanggung oleh asuransi kesehatan, variabel *smoker* mempresentasikan penerima asuransi merokok atau tidak, variabel *region* mempresentasikan wilayah tempat tinggal penerima asuransi di Amerika Serikat, dan variabel *charges* mempresentasikan biaya pengobatan perorangan yang ditagihkan oleh asuransi kesehatan.

```
[1]  import pandas as pd
     from functools import wraps
```

Gambar 1. *Import* Dataset

Pada Gambar 1. *library* yang digunakan aplikasi ini adalah *library pandas* dan fungsi *wraps* dari *functools*. *Pandas* digunakan untuk memanipulasi dan menganalisis data dan *wraps* digunakan untuk membantu pembuatan fungsi dekorator normalisasi data.

```
[2]  def normalize_data(func):

       @wraps(func)
       def decorated_function(df, *args, **kwargs):

         data = df.select_dtypes(include='number')
         normalized_data = (data - data.min()) / (data.max() - data.min())
         df.update(normalized_data)
         return func(df)

       return decorated_function
```

Gambar 2. Fungsi Dekorator Normalisasi Data

Selanjutnya buatlah fungsi dekorator normalisasi data seperti pada Gambar 2. Pada kode tersebut digunakan untuk normalisasi data dengan rumus data dikurangi data terkecil lalu dibagi dengan data terbesar dikurangi data terkecil sehingga nilai normalisasi data akan berada pada rentang nol dan satu, normalisasi ini digunakan sebelum meneruskannya ke fungsi yang diberikan sebagai argumen func.

```
[3]  df = pd.read_csv("insurance.csv")
```

Gambar 3. *Import* Dataset

Setelah membuat fungsi *normalized_data*, langkah selanjutnya adalah *import* dataset asuransi kesehatan yang didefinisikan sebagai *df* dengan menggunakan fungsi *read_csv("insurance.csv")*.

[4]

	age	sex	bmi	children	smoker	region	charges
1051	64	male	26.41	0	no	northeast	14394.5579
651	53	female	39.60	1	no	southeast	10579.7110
979	36	female	29.92	0	no	southeast	4889.0368
589	38	female	30.69	1	no	southeast	5976.8311
803	18	female	42.24	0	yes	southeast	38792.6856

Gambar 4. *Sample* Dataset

Pada Gambar 4. tersebut merupakan lima sampel data berasal dari dataset *insurance* yang telah berhasil di-*import* sebelumnya.

```
[5]  @normalize_data
     def dataset(data):
       return data.sample(5)

     dataset(df)
```

Gambar 5. Fungsi Menampilkan *Sample* Dataset dengan Decorator Normalisasi Data

Setelah dataset berhasil di-*import,* langkah selanjutnya adalah membuat fungsi *dataset* untuk menampilkan data *insurance* yang telah dinormalisasi. Pada Gambar 5. ini menggunakan dekorator @normalize_data pada fungsi *dataset*, yang akan normalisasi data numerik dalam *Data Frame df* sebelum meneruskannya ke fungsi *dataset*. Data numerik dalam *df* dinormalisasi sehingga setiap kolom memiliki nilai antara 0 dan 1. Hal ini memastikan bahwa dataset menerima data yang telah diproses dan berada dalam rentang nilai yang sama. Lalu outputnya berupa Gambar 6. di bawah ini.

[5]

	age	sex	bmi	children	smoker	region	charges
480	0.978261	male	0.682405	0.6	no	northwest	0.230385
270	0.000000	male	0.360775	0.2	no	southeast	0.009538
544	0.782609	male	0.383374	0.0	no	northwest	0.145408
374	0.043478	male	0.467312	0.0	no	southeast	0.004304
1119	0.260870	female	0.107345	0.6	no	northwest	0.072971

Gambar 6. *Sample* Dataset Setelah di Normalisasi

Pada Gambar 6. di atas merupakan tabel yang berisikan lima sampel data yang sudah dinormalisasi dari dataset *insurance.*

```
[6]  @normalize_data
     def statistika_deskriptif(data):
       return data.describe()

     statistika_deskriptif(df)
```

Gambar 7. Fungsi Statistika Deskriptif dengan Decorator Normalisasi Data

Selanjutnya, langkah yang dilakukan adalah membuat fungsi statistika deskriptif seperti pada Gambar 7. untuk menampilkan statistika deskriptif dari dataset *insurance* yang telah dinormalisasi dan ditampilkan kembali dalam bentuk tabel seperti pada Gambar 8. di bawah ini dengan menggunakan fungsi *describe()* dari *library pandas.*

[6]

	age	bmi	children	charges
count	1338.000000	1338.000000	1338.000000	1338.000000
mean	0.461022	0.395572	0.218984	0.193916
std	0.305434	0.164062	0.241099	0.193301
min	0.000000	0.000000	0.000000	0.000000
25%	0.195652	0.278080	0.000000	0.057757
50%	0.456522	0.388485	0.200000	0.131849
75%	0.717391	0.504002	0.400000	0.247700
max	1.000000	1.000000	1.000000	1.000000

Gambar 8. Statistika Deskriptif Dataset Setelah di Normalisasi

Pada Gambar 8. di atas adalah tabel statistika deskriptif dari dataset *insurance* yang telah dinormalisasi. Pada dataset *insurance* untuk variabel *age* yang telah dinormalisasi nilai rata-rata adalah 0,461022 dan standar deviasi 0,305434, variabel bmi yang telah dinormalisasi nilai rata-rata adalah 0,395572 dan standar deviasi 0,164062, variabel *children* yang telah dinormalisasi nilai rata-rata adalah 0,218984 dan standar deviasi 0,241099, dan variabel *charges* yang telah dinormalisasi nilai rata-rata adalah 0,193916 dan standar deviasi 0,193301.

4. Kesimpulan

Pada percobaan kali ini, kami melakukan normalisasi data pada dataset *insurance*, kami melakukan metode normalisasi dengan *min-max scaling* yaitu sebuah teknik sederhana dalam normalisasi data yang cara kerjanya setiap nilai pada sebuah variabel dikurangi dengan nilai minimum variabel tersebut, kemudian dibagi dengan rentang nilai atau nilai maksimum dikurangi nilai minimum dari variabel tersebut, sehingga dalam dataset menjadi rentang dari 0 sampai 1, bisa dilihat pada dataset kami nilai sebelum normalisasi data yang sangat bervariasi dan setelah normalisasi data nilai kami sudah menjadi lebih sederhana dari 0 sampai 1.

Faktor normalisasi ditentukan oleh jumlah digit yang mewakili nilai terbesar dalam dataset. Konsepnya relatif mudah diimplementasikan, karena hanya melibatkan pembagian nilai dengan faktor skala yang terkait dengan jumlah digit maksimum dalam dataset. Namun, normalisasi dengan *min-max scaling* kurang fleksibel dalam menangani data yang memiliki rentang nilai yang sangat besar atau data yang tidak memiliki distribusi yang homogen. Selain itu, teknik ini sensitif terhadap *outlier* karena menggunakan nilai maksimum dalam dataset sebagai faktor normalisasi, sehingga perlu dilakukan penyesuaian jika terdapat *outlier* dalam dataset untuk mencegah pengaruh yang berlebihan dari nilai-nilai ekstrem tersebut. Dengan demikian, Normalisasi dengan *min-max scaling* cocok digunakan untuk dataset dengan distribusi relatif homogen dan tidak memiliki nilai yang sangat ekstrem.

5. Daftar Pustaka

Buzulan, P. (2020, June 19). *Syntactic Sugar in Python. Have you ever heard about syntactic… | by Petru Buzulan | Analytics Vidhya*. Medium. Retrieved April 25, 2024, from https://medium.com/analytics-vidhya/syntactic-sugar-in-python-3e61d1ef2bbf

Kahari, M. M. H. (2022). *Penerapan Algoritma Greedy pada Least Recently Used (LRU) untuk Memoization dalam Optimasi Waktu Pemrosesan*. informatika.stei.itb.ac.id. -

Prayitno, Handoko, S., & Nurfana, A. A. H. (2018). PERENCANAAN DAN PEMBUATAN SISTEM INFORMASI WEB NERACA PT POS INDONESIA PROCESSING CENTRE SEMARANG. *ORBITH*, *14*(1), 74. -

Prayogo, N. A. (2022). *Dasar Pemrograman Golang*. -. -