JOBSHEET 11 Pemilihan dan Pembersihan Data

Tujuan

Setelah menyelesaikan tugas latihan pada jobsheet ini, peserta mampu menentukan melakukan pemilihan dan pembersihan data menggunakan Python

Peralatan yang Dibutuhkan

- Google Colab

1.1. Teori Pendukung

Tahap ketiga dalam CRISP-DM adalah Persiapan Data. Dua langkah awal dama proses persiapan data adalah pemilihan dan pembersihan data.

Pemilihan Data

Sebelumnya pada tahap data understanding sudah dilakukan proses pengumpulan data dari berbagai sumber. Selanjutnya dilakukan pemilihan data yang relevan dengan tujuan data mining yang telah ditentukan. Secara umum, terdapat 2 cara pemilihan data:

- Memilih item (baris)

  Pemilihan baris-baris data mungkin dilakukan karena beberapa alasan, misalnya pemilihan data dalam beberapa tahun terakhir maupun sampling data yang harus dilkaukan karena adanya constraint waktu/sumber daya lainnya.
- Memilih atribut (kolom)

  Atribut/feature dipilih sesuai dengan tujuan data mining. Pemilihan atribut dilakukan berdasarkan knowledge dari domain expert. Feature yang tidak mempengaruhi output data mining tidak akan digunakan pada tahap pemodelan.

Pembersihan Data

Langkah pembersihan data merupakan tindak lanjut dari hasil eksplorasi dan analisa kualitas data yang sudah dilakukan pada tahap data understanding. Permasalahan yang disimpulkan dari tahap tersebut dapat diselesaikan dengan cara berikut:

- Correcting

  Jika terdapat data yang salah karena kesalahan input maupun kesalahan pengukuran, maka perlu dilakukan perbaikan. Misalnya jika terdapat data umur dengan nilai 800, kemungkinan nilai yang benar adalah 80. Jika ragu dan tidak ingin melakukan perbaikan, data tersebut dapat di-exclude

- Completing

  Nilai Null atau data yang hilang harus dilengkapi. Hal tersebut dilakukan karena beberapa algoritma (ex: NN Classification) tidak bisa menghandle nilai null sehingga akan terjadi error di iterasi awal. Dua cara yang dilakukan adalah dengan menghapus record atau mengisikan missing values dengan data yang dapat dipertanggungjawabkan. Menghapus data yang memiliki missing values sangat tidak disarankan, apalagi jika jumlah data/record bermissing values persentasenya besar, kecuali jika record tersebut tidak mungkin dilengkapi. Mengisikan missing values adalah pilihan terbaik karena satu record bisa sangat mempengaruhi hasil data mining. Metodologi dasar untuk mengisikan missing values pada variabel kualitatif adalah menggunakan modus, sedangkan pada variabel kuantitatif menggunakan mean, median, atau mean+standard deviasi acak. Metodologi lain yang digunakan adalah dengan menggunakan kriteria yang lebih spesifik. Contoh: usia rata-rata pada class tertentu

- Converting

  Converting yaitu melakukan konversi format tipe data. Biasanya dilakukan encoding data string ke number categorical. Contoh Sex_code (M/F) – diencode menjadi (0/1). Hal ini dilakukan untuk mempermudah perhitungan median/modus. Converting juga dilakukan sebagai solusi masalah coding scheme yang tidak seragam dan metadata yang tidak sesuai.

- Creating

  Creating atau disebut juga dengan Feature Engineering yaitu menggunakan fitur yang ada untuk membuat fitur baru yang mungkin dapat meningkatkan performa data mining. Selain penambahan fitur, creating juga dapat dilakukan dengan membangun record atau row data baru. Proses creating data akan dibahas pada pertemuan selanjutnya.

1.2. Praktikum

Load dataset ke Google Colab

Load dataset ke google colab dapat dilakukan dengan beberapa cara, di antaranya melalui github repository

- Download file mtcars.csv melalui link https://www.kaggle.com/datasets/ruiromanini/mtcars
- Upload file tersebut ke dalam salah satu github repository
- Klik pada file yang sudah terupload kemudian pilih View Raw. Copy link url dari browser.
- Buat new notebook kemudian tuliskan code berikut

```
import pandas as pd
url="https://raw.githubusercontent.com/zuraidagit/datascience/main/mtcars.csv"
mtcars = pd.read_csv(url)
mtcars.head(10)
```

Library pandas digunakan untuk mengelola dataset, termasuk melakukan analisa, pembersihan, eksplorasi, dan manipulasi data. Isikan variable url dengan url raw data mtcars.csv yang telah diupload ke github repository. Gunakan function read_csv() untuk membaca dataset dari file CSV dan function head(n) untuk mengembalikan sejumlah n baris pertama dari dataframe. Nilai default n adalah 5.

	model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
0	Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
1	Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
2	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
3	Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
4	Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
5	Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
6	Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
7	Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
8	Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
9	Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4

Eksplorasi Data

Lakukan eksplorasi dataset untuk mengidentifikasi nilai descriptive statistics dari dataset menggunakan python

- Tambahkan atribut **model** sebagai index/label dari dataframe **mtcars** kemudian hapus atribut **model** dengan code berikut

```
mtcars.index = mtcars.model
del mtcars["model"]
mtcars.head()
```

model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2

- Menampilkan nilai mean dari setiap feature

```
mtcars.mean()
```

```
mpg      20.090625
cyl       6.187500
disp    230.721875
hp      146.687500
drat      3.596563
wt        3.217250
qsec     17.848750
vs        0.437500
am        0.406250
gear      3.687500
carb      2.812500
dtype: float64
```

- Menampilkan nilai mean dari setiap row

```
mtcars.mean(axis=1)
```

```
model
Mazda RX4              29.907273
Mazda RX4 Wag          29.981364
Datsun 710             23.598182
Hornet 4 Drive         38.739545
Hornet Sportabout      53.664545
Valiant                35.049091
Duster 360             59.720000
Merc 240D              24.634545
Merc 230               27.233636
Merc 280               31.860000
Merc 280C              31.787273
```

- Menampilkan nilai median

```
mtcars.median()
```

```
mpg      19.200
cyl       6.000
disp    196.300
hp      123.000
drat      3.695
wt        3.325
qsec     17.710
vs        0.000
am        0.000
gear      4.000
carb      2.000
dtype: float64
```

- Menampilkan nilai modus

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
0	10.4	8.0	275.8	110.0	3.07	3.44	17.02	0.0	0.0	3.0	2.0
1	15.2	NaN	NaN	175.0	3.92	NaN	18.90	NaN	NaN	NaN	4.0
2	19.2	NaN	NaN	180.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	21.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	21.4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
5	22.8	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
6	30.4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

- Menampilkan nilai Q1

```
mtcars.quantile(0.25)
```

```
mpg      15.42500
cyl       4.00000
disp    120.82500
hp       96.50000
drat      3.08000
wt        2.58125
qsec     16.89250
vs        0.00000
am        0.00000
gear      3.00000
carb      2.00000
Name: 0.25, dtype: float64
```

- Menampilkan nilai Q3 untuk atribut **gear**

```
mtcars["gear"].quantile(0.75)
```

```
4.0
```

- Menampilkan nilai variance dan standar deviasi untuk atribut **mpg**

```
mtcars["mpg"].var()
```

```
36.32410282258064
```

```
mtcars["mpg"].std()
```

```
6.026948052089104
```

Pemilihan Baris

- Pemilihan dengan vectorization

```
mtcars = mtcars[mtcars["mpg"] >= 11]
```

- Pemilihan dengan multiple condition

```
mtcars = mtcars[(mtcars["mpg"] >= 11) & (mtcars["disp"] <= 450)]
```

- Pemilihan dengan loc property

```
mtcars = mtcars.loc[mtcars["mpg"] >= 11]
```

- Pemilihan dengan fungsi query()

```
mtcars = mtcars.query("mpg >= 11")
```

- Penghapusan row dengan fungsi drop()

```
mtcars.drop(mtcars[mtcars["mpg"] < 11].index, inplace = True)
```

Pemilihan kolom

- Pemilihan baris dan kolom sekaligus menggunakan fungsi query()

```
mtcars = mtcars.query("mpg >= 11")[["mpg", "cyl", "disp", "hp", "drat", "wt", "carb"]]
mtcars.head()
```

- Pemilihan kolom menggunakan fungsi filter()

```
mtcars = mtcars.filter(items=["mpg", "cyl", "disp", "hp", "drat"])
mtcars.head()
```

- Penghapusan kolom menggunakan fungsi drop()

```
dropped_column = ["hp", "drat"]
mtcars.drop(dropped_column, axis = 1, inplace = True)
mtcars.head()
```

Pembersihan Data

Download dataset train.csv dari case titanic melalui link berikut dan save as titanic.csv

https://www.kaggle.com/competitions/titanic/data

Upload dataset tersebut ke github, view as raw data, lalu copy url nya

Buat new notebook di google colab kemudian load data dari titanic.csv

```
import pandas as pd
url="https://raw.githubusercontent.com/zuraidagit/datascience/main/titanic.csv"
titanic = pd.read_csv(url)
titanic.head(10)
```

- Mengidentifikasi jumlah nilai Nan pada setiap kolom

```
nan_count = titanic.isna().sum()
print(nan_count)
```

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

- Mengidentifikasi jumlah nilai Nan pada kolom tertentu

```
age_nan_count = titanic["Age"].isna().sum()
print(age_nan_count)
```

```
177
```

- Replace nilai NaN dengan nilai median

```
titanic["Age"].fillna(titanic["Age"].median(), inplace = True)
titanic.head(10)
```

- Replace nilai NaN dengan nilai modus

```
titanic["Embarked"].fillna(titanic["Embarked"].mode()[0], inplace = True)
titanic.head(10)
```

- Melakukan konversi nilai pada kolom Sex menjadi 1 dan 0 dengan fungsi replace()

```
titanic["Sex"].replace("female", 0, inplace = True)
titanic["Sex"].replace("male", 1, inplace = True)
titanic.head()
```

Konversi dengan fungsi replace() juga dapat dilakukan sekaligus dengan dictionary

```
titanic["Sex"].replace({"female": 0, "male": 1}, inplace = True)
titanic.head()
```

- Penghapusan baris dengan kolom bernilai NaN

```
titanic.dropna(subset = ["Cabin"], inplace=True)
titanic.head(10)
```

Menghapus data yang memiliki missing values sangat tidak disarankan, apalagi jika jumlah data/record bermissing values persentasenya besar, kecuali jika record tersebut tidak mungkin dilengkapi. Contoh di atas sebaiknya ***tidak*** *dilakukan karena data dengan nilai NaN pada kolom Cabin ada sejumlah 687 data.*

1.3. Latihan

1. Pilih salah satu dataset dari Kaggle.com
2. Identifikasi nilai mean, median, Q1, Q3, modus, variance, dan standar deviasi dari masing-masing fitur/atribut pada dataset
3. Identifikasi masalah yang ada pada dataset kemudian lakukan solusi pembersihan data yang menurut Anda paling sesuai