

PROGRAM STUDI SARJANA SISTEM INFORMASI
LAPORAN AKHIR PROYEK AKHIR MATA KULIAH
12S4054 - DATA MINING



Fraud Detection Using SVM Algorithm

OLEH:

| | |
|-----------------|---------------------------------------|
| 12S17009 | Prince Ephraim Prabowo Silaban |
| 12S17043 | Enjelin Ida Hutahaean |
| 12S18004 | Rosalia Pane |
| 12S18017 | Putri Yohana Panjaitan |

PROGRAM STUDI SARJANA SISTEM INFORMASI
FAKULTAS INFORMATIKA DAN TEKNIK ELEKTRO
INSTITUT TEKNOLOGI DEL

2021

DAFTAR ISI

BAB 1

| | |
|-----------------------------------|---|
| BUSINESS UNDERSTANDING | 6 |
| 1.1 Determine Business Objectives | 6 |
| 1.2 Access the Situation | 6 |
| 1.3 Determine Data Mining Goals | 7 |
| 1.4 Produce Project Plan | 7 |

BAB 2

| | |
|---------------------------|----|
| DATA UNDERSTANDING | 9 |
| 2.1 Collect Initial Data | 9 |
| 2.2 Describe Data | 9 |
| 2.3 Explore Data | 11 |
| 2.4 Verify Data Quality | 15 |

BAB 3

| | |
|-------------------------|----|
| DATA PREPARATION | 16 |
| 3.1 Package | 16 |
| 3.2 Dataset Description | 16 |
| 3.3 Clean Data | 18 |
| 3.4 Transforming Data | 21 |
| 3.5 Feature Selection | 22 |

BAB 4

| | |
|----------------------------------|----|
| MODELING | 25 |
| 4.1 Selection Modeling Technique | 25 |
| 4.1.1 Modeling Techniques | 25 |
| 4.1.2 Modeling Assumptions | 26 |
| 4.2.1 Test Design | 27 |
| 4.3 Build Model | 27 |
| 4.3.1 Parameter Settings | 27 |
| 4.3.2 Models | 27 |

| | |
|-------------------------------------|----|
| 4.4 Assess Model | 27 |
| BAB 5 | |
| Evaluation | 28 |
| 5.1 Evaluate Result | 28 |
| 5.2 Evaluate Process | 31 |
| 5.3 Determine Next Steps | 31 |
| BAB 6 | |
| DEPLOYMENT | 33 |
| 6.1 Plan Deployment | 33 |
| 6.2 Plan Monitoring and Maintenance | 33 |
| 6.3 Produce Final Report | 34 |
| 6.4 Review Project | 34 |
| DAFTAR PUSTAKA | 35 |

DAFTAR GAMBAR

| | |
|---|----|
| Gambar 1 Fungsi Info | 14 |
| Gambar 2 Fungsi .describe() | 14 |
| Gambar 3 Fungsi .data.corr() | 15 |
| Gambar 4 Fungsi .head() | 15 |
| Gambar 5 Fungsi data.isnull().sum() | 16 |
| Gambar 6 Fungsi .dropna() | 17 |
| Gambar 7 Transforming Data | 17 |
| Gambar 8 SVM dengan kernel linear | 21 |
| Gambar 9 Hasil menggunakan kernel linear | 22 |
| Gambar 10 kernel Polynomial | 22 |
| Gambar 11 Hasil menggunakan kernel polynomial | 22 |
| Gambar 12 Menggunakan kernel Sigmoid | 23 |
| Gambar 13 Hasil menggunakan kernel sigmoid | 23 |
| Gambar 14 SVM dengan kernel RBF | 23 |
| Gambar 15 Hasil menggunakan kernel RBF | 24 |

DAFTAR TABEL

| | |
|---|---|
| Tabel 1 Perencanaan Proyek | 5 |
| Tabel 2 Deskripsi dataset fraud_detection_train.csv | 7 |

BAB 1

BUSINESS UNDERSTANDING

Business Understanding merupakan tahap awal atau tahap pemahaman dalam penelitian. Pada tahap ini dibutuhkan pemahaman mengenai substansi dari kegiatan *data mining* yang akan dilakukan serta kebutuhan dari sebuah perspektif bisnis. Pada tahap *Business Understanding* juga diperlukan pemahaman tentang latar belakang dan tujuan pada proses bisnis yang berhubungan dengan *Fraud Detection*.

1.1 Determine Business Objectives

Pada tahap *Determine Business Objectives*, dijelaskan tujuan bisnis untuk menentukan faktor-faktor penting dalam penelitian yang direncanakan dan memastikan bahwa hasil akhir dari penelitian sesuai dengan yang diharapkan. Semakin berkembangnya teknologi maka semakin banyak informasi yang tersedia. Informasi dapat diakses dengan mudah melalui penggunaan teknologi yang dikaji agar lebih efisien dan optimal melalui internet. *Business Objectives* dari penelitian ini adalah melakukan *Fraud Detection* dengan menggunakan data dari BPJS Hackathon. Setiap atribut yang terdapat pada data BPJS Hackathon akan dianalisis dan dilakukan pemodelan dengan menggunakan algoritma Support Vector Machine (SVM) menggunakan bahasa pemrograman Python.

1.2 Access the Situation

Proyek ini akan melibatkan pencarian fakta yang lebih rinci untuk semua sumber daya (*sources*) seperti sumber daya perangkat keras, sumber daya data (*data sources*) dan sumber daya personal.

1. *Data sources* yang digunakan pada proyek ini adalah dataset *Fraud Detection train* pada studi kasus BPJS Hackathon.
2. Sumber daya perangkat keras yang digunakan pada proyek ini adalah laptop IdeaPad Lenovo 4GB RAM, Processor Intel Core i5-7200U Dual Core 2.5 GHZ Turbo Boost 3.1 GHZ, CD/DVD ROM Drive.
3. Sumber daya personal pada proyek ini terdiri dari 4 orang mahasiswa yang berperan pada pengerjaan proyek mulai dari tahap *business understanding*, *data understanding*, *data preparation*, *modeling*, *evaluation*, dan *deployment*.

1.3 Determine Data Mining Goals


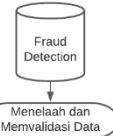
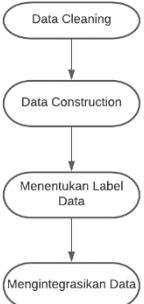
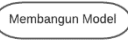
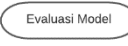
Determine Data Mining Goals adalah tahapan untuk mengubah pengetahuan pada domain bisnis menjadi sebuah *definisi problem data mining* serta untuk menetapkan tujuan *data mining*. Tujuan data mining dalam pengerjaan proyek ini adalah untuk menggali *Discovering Knowledge* mengenai pola (*pattern*) item mengenai *Fraud Detection* menggunakan dataset BPJS Hackathon.

1.4 Produce Project Plan

Tahapan yang dilakukan disini adalah memaparkan rancangan kerja yang ditujukan untuk mencapai tujuan dari data mining sehingga mampu untuk mencapai tujuan bisnis, kemudian menentukan teknik dan *tools* yang selanjutnya akan dipergunakan.

Project plan proyek yaitu untuk dapat menyelesaikan tujuan data mining serta mencapai tujuan bisnis adalah sebagai berikut :

Tabel 1 Perencanaan Proyek

| Business Understanding | Data Understanding | Data Preparation | Modeling | Model Evaluation | Hasil |
|---|--|--|--|--|--|
|  Membuat objective dan tujuan proyek ↓ Membuat Rencana Proyek |  Fraud Detection ↓ Menelaah dan Memvalidasi Data • Rata2 umur pasien berkelamin perempuan yang melakukan kecurangan adalah 38 tahun dan laki laki 36 tahun • pasien rentang umur 50-60 tahun adalah pasien yang paling banyak melakukan kecurangan penggunaan dana BPJS • Terdiri dari 49 fitur numerik dan 5 fitur kategorikal • tidak ditemukan data null maupun data NaN, sehingga tidak perlu dilakukan Handling Missing data • Visualisasi: BarChart dan Heatmap |  Data Cleaning ↓ Data Construction ↓ Menentukan Label Data ↓ Mengintegrasikan Data • Dropping Unecessary Column: beberapa fitur yang tidak penting telah dihapus dari dataset • Transforming Categorical Data: Fitur-fitur kategorikal telah di transformasi menjadi data numeric • Features Selection menggunakan KBest: mendrop fitur yang memiliki satu digit score, dikarenakan fitur2 tersebut tidak akan berdampak banyak pada sahat pemodelan • Scaling Dataset: data yang ada pada dataset memiliki skala yang sama • Split Dataset |  Membangun Model • Support Vector Machine • model pertama menggunakan kernel RBF: 0.5648 |  Evaluasi Model • Model accuracy kernel Linear : 0.5448 • Model accuracy kernel Polynomial : 0.5459 • Model accuracy kernel Sigmoid : 0.5088 • Model accuracy kernel RBF : 0.5782 | • Accuracy 58% • Precision 58% • recall 59%. |

Dalam pelaksanaan proyek dalam penelitian ini, diperlukan *tools data mining* yang mendukung metode untuk berbagai tahapan proses. *Tools* dan teknik yang digunakan dapat mempengaruhi keseluruhan proyek. *Tools* yang digunakan dalam mengerjakan proyek ini adalah *python*. *Python* adalah bahasa pemrograman berorientasi objek yang digunakan

dalam pengembangan perangkat lunak maupun dalam analisis dan *data science*. *Python* memiliki berbagai *library* yang menyediakan fungsi untuk melakukan analisis data, memproses data, memvisualisasikan data, dll.

Python menyediakan *library* seperti *scikit-learn*, Keras, TensorFlow untuk membantu dalam pembuatan model *data mining* dengan cepat. Selain itu, terdapat juga *library* yang dapat digunakan untuk membagi *dataset* menjadi data *training* dan data *test*, misalnya menggunakan *cross-validation*. Metode atau algoritma yang akan digunakan dalam proyek ini adalah algoritma Support Vector Machine (SVM) yang termasuk dalam *Supervised Learning* pada penambangan data (*Data Mining*).

BAB 2

DATA UNDERSTANDING

Data Understanding atau pemahaman data merupakan tahap pengumpulan data awal dan meneliti data yang bertujuan untuk mengidentifikasi dan mempelajari data untuk bisa mengenal data yang akan dipakai. Tahap ini mencoba mengidentifikasikan masalah yang berkaitan dengan kualitas data, mendeteksi subset yang menarik dari data untuk membuat hipotesa awal.

2.1 Collect Initial Data

Collect Initial Data adalah proses pengumpulan data untuk dapat digunakan, data dapat diperoleh dengan melakukan kuesioner, wawancara, mengambil langsung sampel data dari lapangan, maupun dari internet. Penulis akan menggunakan data *Fraud Detection train* pada studi kasus BPJS Hackathon.

Berikut adalah data yang akan digunakan pada pengerjaan proyek ini, yaitu data *Fraud Detection train* pada studi kasus BPJS Hackathon.

```
#import library
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

data = pd.read_csv("dataset/bpjs.csv")

data.sample(10)
```

| | visit_id | kdkc | dati2 | typeppk | jkpst | umur | jnspelsep | los | cmg | severitylevel | ... | proc63_67 | proc68_70 | proc71_73 | proc74_75 | proc76_77 | proc78_79 | proc80_99 | proc... |
|--------|----------|------|-------|---------|-------|------|-----------|-----|-----|---------------|-----|-----------|-----------|-----------|-----------|-----------|-----------|-----------|---------|
| 66033 | 66034 | 401 | 69 | SC | L | 36 | 1 | 2 | N | 1 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 3 | |
| 15043 | 15044 | 2302 | 346 | C | P | 48 | 2 | 0 | I | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 3 | |
| 58511 | 58512 | 1108 | 171 | GD | L | 17 | 2 | 0 | Q | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 131329 | 131330 | 2201 | 227 | SC | P | 50 | 1 | 2 | L | 1 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 1 | |
| 97977 | 97978 | 1314 | 195 | D | P | 33 | 1 | 2 | B | 2 | ... | 0 | 0 | 1 | 0 | 0 | 0 | 0 | |
| 73740 | 73741 | 203 | 31 | C | P | 54 | 2 | 0 | Q | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 27373 | 27374 | 601 | 90 | I3 | P | 73 | 2 | 0 | Q | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 176783 | 176784 | 501 | 82 | C | P | 57 | 2 | 0 | G | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 30944 | 30945 | 1107 | 150 | SB | L | 0 | 1 | 2 | P | 1 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 120480 | 120481 | 2103 | 308 | C | P | 47 | 1 | 3 | E | 1 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

10 rows × 53 columns

2.2 Describe Data

Pada tahap *describe the data*, penulis akan memahami karakteristik dari data dengan menganalisa setiap atribut yang berada di dalam data, termasuk juga melakukan analisis apakah ada objek data yang bersifat *noisy*. Hal ini dilakukan untuk memperoleh informasi

terkait data yang akan digunakan. Pada tahap ini, penulis akan melakukan *exploratory data analysis* (EDA) untuk memahami karakteristik dari data. Deskripsi data bertujuan untuk menggambarkan data agar lebih mudah dipahami dan dimengerti oleh penulis atau pembaca terkait dengan proyek yang dibangun.

Adapun deskripsi dari data yang akan digunakan dapat dilihat pada tabel berikut.

Tabel 2 Deskripsi dataset `fraud_detection_train.csv`

| <i>Variable name</i> | Tipe Atribut | <i>Variable description</i> |
|------------------------------|---------------------|---|
| visit_id | numerik | id kunjungan |
| kdkc | numerik | Kode wilayah kantor cabang BPJS Kesehatan |
| dati2 | numerik | Kode kabupaten/kota |
| typeppk | kategorikal | Kode tipe dari rumah sakit |
| jkpst | kategorikal | Jenis kelamin peserta JKN-KIS |
| umur | numerik | Umur peserta saat mendapatkan pelayanan rumah sakit |
| jnspelsep | numerik | Tingkat pelayanan: 1.rawat inap; 2.rawat jalan; |
| los | numerik | Lama peserta dirawat di rumah sakit |
| cmg | kategorikal | Klasifikasi CMG (Case Mix Group) |
| severitylevel | numerik | Tingkat urgensi (0 dan 1) |
| diagprimer | kategorikal | Diagnosa primer |
| dx2_a00_b99 - dx2_z00_z99 | numerik | Diagnosa sekunder |
| proc00_13 – procv00_v89 | numerik | Kode kelompok procedure |
| label | numerik | Flag fraud: 1:fraud; 0:tidak fraud |

2.3 Explore Data

Pada tahap *Exploratory Data Analysis* (EDA) diperlukan sebagai sebuah pendekatan dalam menganalisis dataset untuk meringkas karakteristik utama *dataset*. Biasanya dilakukan dengan menggunakan metode visual. EDA digunakan untuk memahami data, mendapatkan konteks data, memahami variabel dan hubungan di antara variabel, dan merumuskan hipotesis yang berguna dalam membangun model prediksi. Atribut atau fitur pada dataset tidak semua diperlukan dalam menganalisis. Eksplorasi data juga memperhatikan ekstensi dari data yang akan digunakan. Oleh karena itu eksplorasi data pada penelitian ini akan dilakukan dengan melakukan analisis terhadap dimensi dari data yang digunakan, termasuk mengelompokkan data berdasarkan variabel target.

Pembagian Data

1. Pada pengerjaan proyek ini, dilakukan pembagian data menjadi dua berdasarkan label, yaitu fraud dan non-fraud.

```
#Membagi data menjadi menjadi 2 berdasarkan Label (fraud/non-fraud)
grouped = data.groupby("label")

data_fraud = grouped.get_group(1)
data_non_fraud = grouped.get_group(0)

data_fraud.sample(5)
```

| | visit_id | kdkc | dati2 | typeppk | jkpst | umur | jnspelsep | los | cmg | severitylevel | ... | proc63_67 | proc68_70 | proc71_73 | proc74_75 | proc76_77 | proc78_79 | proc80_99 | proc |
|-------|----------|------|-------|---------|-------|------|-----------|-----|-----|---------------|-----|-----------|-----------|-----------|-----------|-----------|-----------|-----------|------|
| 83423 | 83424 | 1005 | 134 | B | P | 28 | 2 | 0 | Q | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 55673 | 55674 | 701 | 97 | SC | P | 7 | 2 | 0 | U | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 82499 | 82500 | 1108 | 171 | C | L | 52 | 2 | 0 | Q | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 21961 | 21962 | 1701 | 290 | B | L | 0 | 1 | 3 | P | 1 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10657 | 10658 | 1603 | 297 | C | L | 52 | 2 | 0 | H | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |

5 rows × 53 columns

2. Untuk melihat rata-rata umur pasien yang melakukan kecurangan berdasarkan gender

```
#melihat rata2 umur pasien yang melakukan kecurangan berdasarkan gender
data_fraud.groupby('jkpst', as_index=False).umur.mean()
```

| | jkpst | umur |
|---|-------|-----------|
| 0 | L | 36.330913 |
| 1 | P | 37.893959 |

Kita dapat melihat bahwa Output dari kode program diatas rata-rata umur pasien berkelamin perempuan yang melakukan kecurangan adalah 38 tahun dan laki laki 36 tahun.

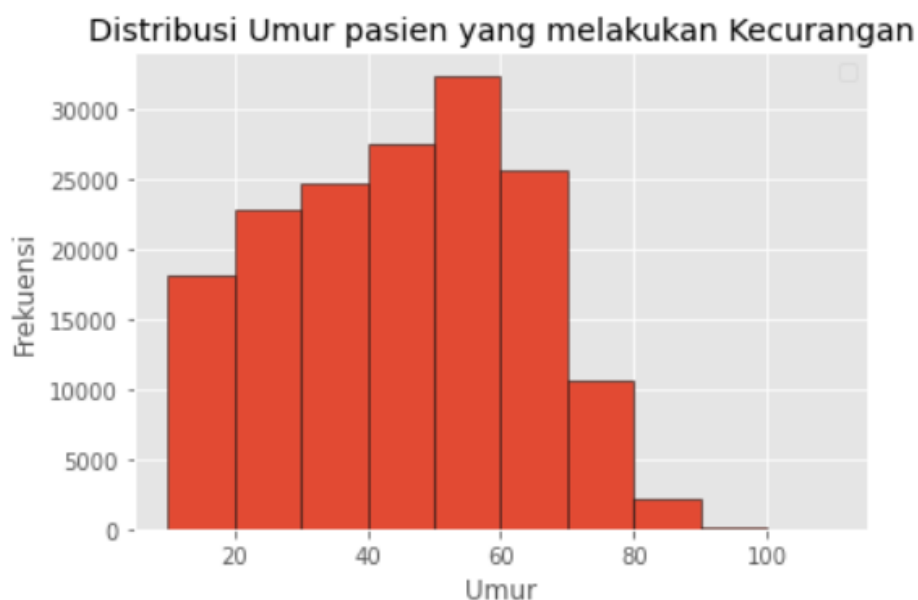
3. Untuk melihat deskripsi data pasien yang melakukan kecurangan mengenai umur

```
#melihat deskripsi data umur pasien yang curang  
data['umur'].describe()
```

```
count      200217.000000  
mean        36.850602  
std         23.095928  
min          0.000000  
25%         18.000000  
50%         39.000000  
75%         56.000000  
max        109.000000  
Name: umur, dtype: float64
```

```
import matplotlib.pyplot as plt  
  
plt.style.use('ggplot')  
ages = data['umur']  
bins = [10, 20, 30, 40, 50, 60, 70, 80, 90,100,110]  
  
plt.hist(ages, bins=bins, edgecolor='black')  
plt.xlabel("Umur")  
plt.ylabel("Frekuensi")  
plt.title("Distribusi Umur pasien yang melakukan Kecurangan")  
plt.legend()  
plt.show()
```

No handles with labels found to put in legend.



Dari kode program diatas dan histogram diatas maka didapatkan keluaran bahwa pasien rentang umur 50-60 tahun adalah pasien yang paling banyak melakukan kecurangan penggunaan dana BPJS.

4. Info Data, pada bagian ini dilakukan untuk mengetahui fitur yang terdapat pada data tersebut. Berikut adalah fitur yang terdapat pada data tersebut

```
#check info dataset
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200217 entries, 0 to 200216
Data columns (total 53 columns):
#   Column                Non-Null Count  Dtype
---  -
0   visit_id              200217 non-null  int64
1   kdkc                  200217 non-null  int64
2   dati2                 200217 non-null  int64
3   typeppk               200217 non-null  object
4   jkpst                 200217 non-null  object
5   umur                  200217 non-null  int64
6   jnspelsep             200217 non-null  int64
7   los                   200217 non-null  int64
8   cmg                   200217 non-null  object
9   severitylevel         200217 non-null  int64
10  diagprimer            200217 non-null  object
11  dx2_a00_b99           200217 non-null  int64
12  dx2_c00_d48           200217 non-null  int64
13  dx2_d50_d89           200217 non-null  int64
14  dx2_e00_e90           200217 non-null  int64
15  dx2_f00_f99           200217 non-null  int64
16  dx2_g00_g99           200217 non-null  int64
17  dx2_h00_h59           200217 non-null  int64
18  dx2_h60_h95           200217 non-null  int64
19  dx2_i00_i99           200217 non-null  int64
20  dx2_j00_j99           200217 non-null  int64
21  dx2_k00_k93           200217 non-null  int64
22  dx2_l00_l99           200217 non-null  int64
23  dx2_m00_m99           200217 non-null  int64
24  dx2_n00_n99           200217 non-null  int64
25  dx2_o00_o99           200217 non-null  int64
26  dx2_p00_p96           200217 non-null  int64
27  dx2_q00_q99           200217 non-null  int64
28  dx2_r00_r99           200217 non-null  int64
29  dx2_s00_t98           200217 non-null  int64
..   ..                 ..
```

```

30 dx2_u00_u99      200217 non-null int64
31 dx2_v01_y98      200217 non-null int64
32 dx2_z00_z99      200217 non-null int64
33 proc00_13         200217 non-null int64
34 proc14_23         200217 non-null int64
35 proc24_27         200217 non-null int64
36 proc28_28         200217 non-null int64
37 proc29_31         200217 non-null int64
38 proc_32_38        200217 non-null int64
39 proc39_45         200217 non-null int64
40 proc46_51         200217 non-null int64
41 proc52_57         200217 non-null int64
42 proc58_62         200217 non-null int64
43 proc63_67         200217 non-null int64
44 proc68_70         200217 non-null int64
45 proc71_73         200217 non-null int64
46 proc74_75         200217 non-null int64
47 proc76_77         200217 non-null int64
48 proc78_79         200217 non-null int64
49 proc80_99         200217 non-null int64
50 proce00_e99       200217 non-null int64
51 procv00_v89       200217 non-null int64
52 label            200217 non-null int64
dtypes: int64(49), object(4)
memory usage: 81.0+ MB

```

Dari kode program diatas, didapatkan output bahwa data tersebut terdiri dari 49 int64 dan object 4.

5. Visualisasi korelasi antara fitur, pada bagian ini kita dapat mengetahui korelasi antara fitur pada data, dan fitur mana yang tidak memiliki korelasi dengan fitur lainnya



Dari kode program diatas, maka didapatkan hasil/output bahwa `procv00_v89','dx2_u00_u99','dx2_koo_k93'` tidak memiliki korelasi apapun antara fitur yang lain, sehingga ada baiknya fitur ini di drop.

2.4 Verify Data Quality

Pada tahap *verify data quality*, dilakukan verifikasi terhadap pengerjaan eksplorasi data untuk memastikan tidak ada data yang bersifat *noisy*. Hal ini dilakukan untuk menghindari kesalahan pada tahap pemodelan. Tahap mengevaluasi kualitas data dan kelengkapan data atau nilai-nilai yang hilang sering terjadi, terutama jika data yang dikumpulkan di jangka waktu yang lama. Memeriksa atribut yang hilang atau kosong. Menilai apakah semua nilai masuk akal, ejaan nilai-nilai, dan apakah atribut dengan nilai yang berbeda memiliki arti yang sama.

BAB 3

DATA PREPARATION

Data preparation merupakan tahap setelah dilakukan pengumpulan data awal yang telah dilakukan pada fase sebelumnya, yaitu *business understanding*. Pada tahap *data preparation* ini, dilakukan proses menyiapkan data awal, memilih variabel yang akan dianalisis dan membersihkan data. Dalam pengerjaan proyek, bahasa pemrograman yang digunakan adalah pemrograman *python* dengan *software* pengolah data Jupyter Notebook.

3.1 Package

Untuk dapat menjalankan beberapa kode program yang akan dijalankan, dibutuhkan beberapa *package* yang harus diinstal, yaitu:

1. **Pandas**, untuk memuat sebuah file ke dalam tabel virtual seperti *spreadsheet*, mengumpulkan data, dan mengolahnya.
2. **Numpy**, untuk operasi vektor dan matriks serta analisis data.
3. **Matplotlib**, untuk menyajikan data ke dalam visual yang lebih menarik dan rapi.

3.2 Dataset Description

Pada fase ini, *dataset* akan dideskripsikan dengan memanfaatkan bahasa pemrograman *python*. Berikut beberapa fungsi yang dijalankan dalam mendeskripsikan *dataset* tersebut:

1. **.info()**, untuk menampilkan gambaran mengenai *dataset*.


```

✓ [10] data.info()
0 d
Data columns (total 53 columns):
#   Column                Non-Null Count  Dtype
---  -
0   visit_id              200217 non-null int64
1   kdkc                  200217 non-null int64
2   dati2                 200217 non-null int64
3   typeppk               200217 non-null object
4   jkpst                 200217 non-null object
5   umur                  200217 non-null int64
6   jnspelsep             200217 non-null int64
7   los                   200217 non-null int64
8   cmg                   200217 non-null object
9   severitylevel         200217 non-null int64
10  diagprimer            200217 non-null object
11  dx2_a00_b99           200217 non-null int64
12  dx2_c00_d48           200217 non-null int64
13  dx2_d50_d89           200217 non-null int64
14  dx2_e00_e90           200217 non-null int64
15  dx2_f00_f99           200217 non-null int64
16  dx2_g00_g99           200217 non-null int64
17  dx2_h00_h59           200217 non-null int64
18  dx2_h60_h95           200217 non-null int64
19  dx2_i00_i99           200217 non-null int64
20  dx2_j00_j99           200217 non-null int64
21  dx2_koo_k93           200217 non-null int64
22  dx2_l00_l99           200217 non-null int64
23  dx2_m00_m99           200217 non-null int64
24  dx2_n00_n99           200217 non-null int64

```

Gambar 1 Fungsi Info

2. **.describe()**, untuk menampilkan berbagai ringkasan atau deskripsi statistik data, seperti jumlah data di setiap kolom (count), rata-rata nilai per kolom (mean), standar deviasi (std), nilai minimum (min), nilai maksimum (max), serta batas nilai dari masing-masing kuartil (25%, 50%, 75%). Berikut beberapa ringkasan atau deskripsi statistik data pada atribut yang bertipe data numerik.

```

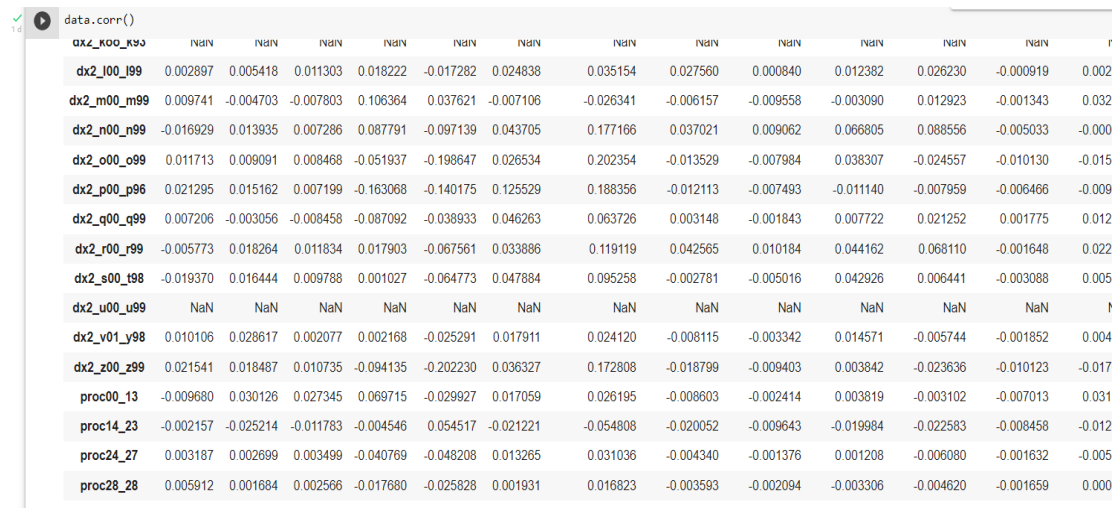
✓ [11] data.describe()
1 d

```

| | visit_id | kdkc | dati2 | umur | jnspelsep | los | severitylevel | dx2_a00_b99 | dx2_c00_d48 | dx2_d50_d89 | dx2_e00_e90 |
|-------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
| count | 200217.000000 | 200217.000000 | 200217.000000 | 200217.000000 | 200217.000000 | 200217.000000 | 200217.000000 | 200217.000000 | 200217.000000 | 200217.000000 | 200217.000000 |
| mean | 100109.000000 | 1147.367816 | 184.793309 | 36.850602 | 1.669778 | 1.303356 | 0.444003 | 0.024893 | 0.008341 | 0.020703 | 0.048213 |
| std | 57797.813761 | 574.486224 | 107.226676 | 23.095928 | 0.470294 | 5.639751 | 0.725227 | 0.162484 | 0.093386 | 0.146842 | 0.244711 |
| min | 1.000000 | 101.000000 | 1.000000 | 0.000000 | 1.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 25% | 50055.000000 | 903.000000 | 114.000000 | 18.000000 | 1.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 50% | 100109.000000 | 1101.000000 | 169.000000 | 39.000000 | 2.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 75% | 150163.000000 | 1314.000000 | 232.000000 | 56.000000 | 2.000000 | 2.000000 | 1.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| max | 200217.000000 | 2606.000000 | 528.000000 | 109.000000 | 2.000000 | 592.000000 | 3.000000 | 4.000000 | 3.000000 | 3.000000 | 7.000000 |

Gambar 2 Fungsi .describe()

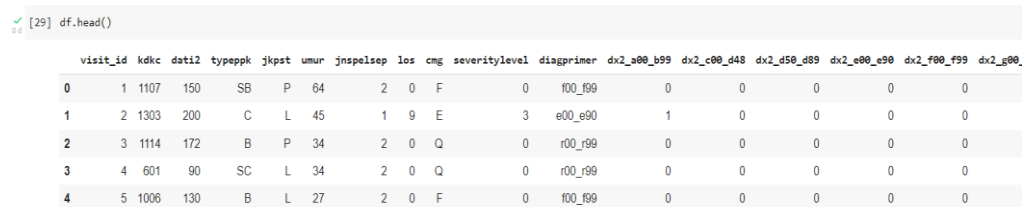
3. `.data.corr()`, untuk melakukan korelasi berpasangan



| | dx2_k00_r99 | dx2_l00_l99 | dx2_m00_m99 | dx2_n00_n99 | dx2_o00_o99 | dx2_p00_p99 | dx2_q00_q99 | dx2_r00_r99 | dx2_s00_s99 | dx2_u00_u99 | dx2_v01_y98 | dx2_z00_z99 | proc00_13 | proc14_23 | proc24_27 | proc28_28 |
|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-----------|-----------|-----------|-----------|
| dx2_k00_r99 | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| dx2_l00_l99 | 0.002897 | 0.005418 | 0.011303 | 0.018222 | -0.017282 | 0.024838 | 0.035154 | 0.027560 | 0.000840 | 0.012382 | 0.026230 | -0.000919 | 0.002 | | | |
| dx2_m00_m99 | 0.009741 | -0.004703 | -0.007803 | 0.106364 | 0.037621 | -0.007106 | -0.026341 | -0.006157 | -0.009558 | -0.003090 | 0.012923 | -0.001343 | 0.032 | | | |
| dx2_n00_n99 | -0.016929 | 0.013935 | 0.007286 | 0.087791 | -0.097139 | 0.043705 | 0.177166 | 0.037021 | 0.009062 | 0.066805 | 0.088556 | -0.005033 | -0.000 | | | |
| dx2_o00_o99 | 0.011713 | 0.009091 | 0.008468 | -0.051937 | -0.198647 | 0.026534 | 0.202354 | -0.013529 | -0.007984 | 0.038307 | -0.024557 | -0.010130 | -0.015 | | | |
| dx2_p00_p99 | 0.021295 | 0.015162 | 0.007199 | -0.163068 | -0.140175 | 0.125529 | 0.188356 | -0.012113 | -0.007493 | -0.011140 | -0.007959 | -0.006466 | -0.009 | | | |
| dx2_q00_q99 | 0.007206 | -0.003056 | -0.008458 | -0.087092 | -0.038933 | 0.046263 | 0.063726 | 0.003148 | -0.001843 | 0.007722 | 0.021252 | 0.001775 | 0.012 | | | |
| dx2_r00_r99 | -0.005773 | 0.018264 | 0.011834 | 0.017903 | -0.067561 | 0.033886 | 0.119119 | 0.042565 | 0.010184 | 0.044162 | 0.068110 | -0.001648 | 0.022 | | | |
| dx2_s00_s99 | -0.019370 | 0.016444 | 0.009788 | 0.001027 | -0.064773 | 0.047884 | 0.095258 | -0.002781 | -0.005016 | 0.042926 | 0.006441 | -0.003088 | 0.005 | | | |
| dx2_u00_u99 | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | | | |
| dx2_v01_y98 | 0.010106 | 0.028617 | 0.002077 | 0.002168 | -0.025291 | 0.017911 | 0.024120 | -0.008115 | -0.003342 | 0.014571 | -0.005744 | -0.001852 | 0.004 | | | |
| dx2_z00_z99 | 0.021541 | 0.018487 | 0.010735 | -0.094135 | -0.202230 | 0.036327 | 0.172808 | -0.018799 | -0.009403 | 0.003842 | -0.023636 | -0.010123 | -0.017 | | | |
| proc00_13 | -0.009680 | 0.030126 | 0.027345 | 0.069715 | -0.029927 | 0.017059 | 0.026195 | -0.008603 | -0.002414 | 0.003819 | -0.003102 | -0.007013 | 0.031 | | | |
| proc14_23 | -0.002157 | -0.025214 | -0.011783 | -0.004546 | 0.054517 | -0.021221 | -0.054808 | -0.020052 | -0.009643 | -0.019984 | -0.022583 | -0.008458 | -0.012 | | | |
| proc24_27 | 0.003187 | 0.002699 | 0.003499 | -0.040769 | -0.048208 | 0.013265 | 0.031036 | -0.004340 | -0.001376 | 0.001208 | -0.006080 | -0.001632 | -0.005 | | | |
| proc28_28 | 0.005912 | 0.001684 | 0.002566 | -0.017680 | -0.025828 | 0.001931 | 0.016823 | -0.003593 | -0.002094 | -0.003306 | -0.004620 | -0.001659 | 0.000 | | | |

Gambar 3 Fungsi `.data.corr()`

4. `.head()`, untuk melihat 5 sampel data teratas.



| | visit_id | kdkc | datil2 | typeppk | jkpt | umur | jnspelese | los | cmg | severitylevel | diagprimer | dx2_s00_b99 | dx2_c00_d48 | dx2_d50_d89 | dx2_e00_e90 | dx2_f00_f99 | dx2_g00 |
|---|----------|------|--------|---------|------|------|-----------|-----|-----|---------------|------------|-------------|-------------|-------------|-------------|-------------|---------|
| 0 | 1 | 1107 | 150 | SB | P | 64 | 2 | 0 | F | 0 | r00_r99 | 0 | 0 | 0 | 0 | 0 | |
| 1 | 2 | 1303 | 200 | C | L | 45 | 1 | 9 | E | 3 | e00_e90 | 1 | 0 | 0 | 0 | 0 | |
| 2 | 3 | 1114 | 172 | B | P | 34 | 2 | 0 | Q | 0 | r00_r99 | 0 | 0 | 0 | 0 | 0 | |
| 3 | 4 | 601 | 90 | SC | L | 34 | 2 | 0 | Q | 0 | r00_r99 | 0 | 0 | 0 | 0 | 0 | |
| 4 | 5 | 1006 | 130 | B | L | 27 | 2 | 0 | F | 0 | r00_r99 | 0 | 0 | 0 | 0 | 0 | |

Gambar 4 Fungsi `.head()`

3.3 Clean Data

Pada fase ini dilakukan pembersihan data. Data cleaning yang dilakukan adalah dengan cara menghapus objek data yang tidak mengandung nilai (missing value).

1. Fungsi `data.isnull().sum()`

Fungsi `Isnull ()` berguna untuk memeriksa suatu kolom ada datanya atau tidak. dan jika tidak ada datanya atau NULL, maka diberi data pengganti.

Fungsi `sum()` berguna untuk memudahkan dalam memahami data, maka perlu dilakukan agregasi data dengan menggunakan fungsi `sum()`, maka akan diketahui berapa jumlah data yang missing value dan berasal dari atribut apa.

```
data.isNull().sum()

visit_id      0
kdkc          0
dati2         0
typeppk       0
jkipst        0
umur          0
jnspelsep     0
los           0
cmg           0
severitylevel  0
diagprimer    0
dx2_a00_b99   0
dx2_c00_d48   0
dx2_d50_d89   0
dx2_e00_e90   0
dx2_f00_f99   0
dx2_g00_g99   0
dx2_h00_h59   0
dx2_h60_h95   0
dx2_i00_i99   0
dx2_j00_j99   0
dx2_k00_k93   0
dx2_l00_l99   0
dx2_m00_m99   0
dx2_n00_n99   0
dx2_o00_o99   0
dx2_p00_p99   0
```

Gambar 5 Fungsi data.isNull().sum()

2. Fungsi data.isna().sum()

Fungsi isna() berguna untuk mengembalikan nilai boolean (True dan False). Jika cell berisi value “False”, maka artinya cell tersebut tidak mengandung missing value dan sebaliknya, jika cell berisi value “True”, maka cell tersebut mengandung missing value.

Untuk memudahkan dalam memahami data, maka perlu dilakukan agregasi data dengan fungsi sum(). Dengan menggunakan fungsi sum(), maka akan diketahui berapa jumlah data yang missing value dan berasal dari atribut apa.

```
#checking NaN data  
data.isna().sum()
```

| | |
|---------------|---|
| visit_id | 0 |
| kdkc | 0 |
| dati2 | 0 |
| typeppk | 0 |
| jkpst | 0 |
| umur | 0 |
| jnspelsep | 0 |
| los | 0 |
| cmg | 0 |
| severitylevel | 0 |
| diagprimer | 0 |
| dx2_a00_b99 | 0 |
| dx2_c00_d48 | 0 |
| dx2_d50_d89 | 0 |
| dx2_e00_e90 | 0 |
| dx2_f00_f99 | 0 |
| dx2_g00_g99 | 0 |
| dx2_h00_h59 | 0 |
| dx2_h60_h95 | 0 |
| dx2_i00_i99 | 0 |
| dx2_j00_j99 | 0 |
| dx2_k00_k93 | 0 |
| dx2_l00_l99 | 0 |
| dx2_m00_m99 | 0 |
| dx2_n00_n99 | 0 |
| dx2_o00_o99 | 0 |
| dx2_p00_p96 | 0 |
| dx2_q00_q99 | 0 |
| dx2_r00_r99 | 0 |
| dx2_s00_t98 | 0 |
| dx2_u00_u99 | 0 |
| dx2_v01_y98 | 0 |
| dx2_z00_z99 | 0 |

Gambar 6 Fungsi data.isna().sum()

Setelah dilakukan pengecekan, tidak ditemukan data null maupun data NaN, sehingga tidak perlu dilakukan Handling Missing data.

3. Fungsi .dropna()

fungsi .dropna() untuk menghilangkan data yang hilang. Setelah fungsi .dropna() dijalankan, maka data yang mengandung missing value terhapus.

```

✓ [24] print(ebola_dropna)
0d
      visit_id kdkc  dati2  ...  proce00_e99  procv00_v89  label
0           1  1107   150  ...           0           0        1
1           2  1303   200  ...           0           0        1
2           3  1114   172  ...           0           0        1
3           4   601    90  ...           0           0        1
4           5  1006   130  ...           0           0        1
...         ...   ...   ...  ...         ...         ...        ...
200212    200213  2102   353  ...           0           0        0
200213    200214  1308   212  ...           0           0        0
200214    200215   201    38  ...           0           0        0
200215    200216  1008   128  ...           0           0        0
200216    200217  1016   117  ...           0           0        0

[200217 rows x 53 columns]

```

Gambar 6 Fungsi .dropna()

4. Dropping Unnecessary Column

1. Dropping Unnecessary Column

```

#Drop fitur yang tidak penting
data_new=data.drop(['visit_id','procv00_v89','dx2_u00_u99','dx2_koo_k93'], axis = 1)

```

```
data_new.sample(10)
```

| | kdkc | dati2 | typeppk | jkpst | umur | jnspelsep | los | cmg | severitylevel | diagprimer | ... | proc58_62 | proc63_67 | proc68_70 | proc71_73 | proc74_75 | proc76_77 | proc78_79 | pro |
|--------|------|-------|---------|-------|------|-----------|-----|-----|---------------|------------|-----|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----|
| 147447 | 1004 | 220 | B | L | 0 | 1 | 7 | P | 1 | p00_p96 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 170959 | 101 | 17 | SC | P | 18 | 2 | 0 | Q | 0 | h00_h59 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5469 | 1703 | 287 | C | P | 40 | 2 | 0 | Z | 0 | z00_z99 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 44142 | 302 | 54 | B | L | 59 | 2 | 0 | F | 0 | f00_f99 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 150892 | 1108 | 171 | SB | L | 51 | 1 | 1 | J | 3 | q00_q99 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 177793 | 1312 | 209 | C | P | 68 | 2 | 0 | Q | 0 | i00_i99 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 94191 | 201 | 38 | SB | P | 45 | 2 | 0 | Q | 0 | h60_h95 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 150023 | 903 | 113 | A | P | 59 | 1 | 25 | J | 3 | j00_j99 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 167118 | 1002 | 133 | SC | P | 52 | 2 | 0 | F | 0 | f00_f99 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 51260 | 1307 | 196 | C | L | 60 | 1 | 4 | D | 2 | d50_d89 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

10 rows x 49 columns

Dari Kode program diatas, maka didapatkan hasil bahwa fitur yang tidak penting telah dihapus dari dataset.

3.4 Transforming Data

Untuk meningkatkan efisiensi data mining, maka perlu melakukan transforming data ke dalam bentuk data yang diperlukan. Pada fase transforming data ini, yang dilakukan adalah mentransformasikan setiap data kategorikal.

```
#Untuk mentransformasi data kategorikal tersebut kita menggunakan Label Encoder dari Scikit Learn
from sklearn import preprocessing
lab_enc = preprocessing.LabelEncoder()
```

```
#Mentransformasi setiap data kategorikal
data_new['typeppk'] = lab_enc.fit_transform(data[['typeppk']])
data_new['jkpst'] = lab_enc.fit_transform(data[['jkpst']])
data_new['cmg'] = lab_enc.fit_transform(data[['cmg']])
data_new['diagprimer'] = lab_enc.fit_transform(data[['diagprimer']])
```

C:\Users\Prince Silaban\anaconda3\lib\site-packages\sklearn\utils\validation.py:63: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
return f(*args, **kwargs)

```
data_new.sample(5)
```

| | kdkc | dat2 | typeppk | jkpst | umur | jnspelsep | los | cmg | severitylevel | diagprimer | ... | proc58_62 | proc63_67 | proc68_70 | proc71_73 | proc74_75 | proc76_77 | proc78_79 | pro |
|--------|------|------|---------|-------|------|-----------|-----|-----|---------------|------------|-----|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----|
| 143306 | 2202 | 228 | 23 | 1 | 35 | 1 | 3 | 19 | 1 | 7 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 913 | 1002 | 133 | 9 | 1 | 0 | 1 | 17 | 0 | 2 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 53304 | 1806 | 344 | 9 | 0 | 16 | 2 | 0 | 16 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 127922 | 1114 | 162 | 18 | 0 | 62 | 2 | 0 | 9 | 0 | 8 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 40082 | 1018 | 221 | 23 | 0 | 38 | 1 | 0 | 11 | 1 | 12 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

5 rows × 49 columns

Gambar 7 Transforming Data

3.5 Feature Selection

1. Pada Tahap ini kita akan memilih fitur-fitur yang digunakan untuk dimodelkan menggunakan Select K-Best

```

import pandas as pd
import numpy as np
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2

X=data_new.drop(columns=['label'])
y = data_new['label'].values

#apply SelectKBest class to extract top 10 best features

bestfeatures = SelectKBest(score_func=chi2, k=10)
fit = bestfeatures.fit(X,y)
dfscores = pd.DataFrame(fit.scores_)
dfcolumns = pd.DataFrame(X.columns)

featureScores = pd.concat([dfcolumns,dfscores],axis=1)
featureScores.columns = ['Specs', 'Score']
print(featureScores.nlargest(49, 'Score'))

```

| | Specs | Score |
|----|---------------|-------------|
| 0 | kdkc | 9795.662620 |
| 1 | dati2 | 8669.031951 |
| 6 | los | 7942.436161 |
| 2 | typeppk | 5990.419433 |
| 46 | proc80_99 | 560.776145 |
| 4 | umur | 528.046221 |
| 36 | proc39_45 | 395.828015 |
| 43 | proc74_75 | 332.211238 |
| 24 | dx2_p00_p96 | 323.959052 |
| 37 | proc46_51 | 155.610284 |
| 27 | dx2_s00_t98 | 138.840193 |
| 29 | dx2_z00_z99 | 111.837531 |
| 13 | dx2_e00_e90 | 89.431012 |
| 8 | severitylevel | 79.539135 |
| 5 | jnspelsep | 75.101308 |
| 23 | dx2_o00_o99 | 73.363121 |
| 22 | dx2_n00_n99 | 65.891060 |
| 16 | dx2_h00_h59 | 61.612817 |

Dari hasil kode program diatas, dapat kita lihat pe-rankingan fitur berdasarkan Select K-Best, berdasarkan ranking tersebut kita akan mendrop fitur yang memiliki satu digit score, dikarenakan fitur-fitur tersebut tidak akan berdampak banyak pada saat pemodelan.

1. Pada tahap ini dilakukan Drop Fitur tidak penting

```

#Drop fitur yang tidak penting
data_new=data_new.drop(['cmg','jkipst','dx2_d50_d89','dx2_f00_f99','dx2_l00_l99','proc52_57','proc24_27','dx2_j00_j99','dx2_r00_r99','proc14_23','proc

```

data_new.sample(5)

| | kdkc | dati2 | typeppk | umur | jnspelsep | los | severitylevel | dx2_a00_b99 | dx2_c00_d48 | dx2_e00_e90 | ... | proc39_45 | proc46_51 | proc63_67 | proc71_73 | proc74_75 | proc76 |
|--------|------|-------|---------|------|-----------|-----|---------------|-------------|-------------|-------------|-----|-----------|-----------|-----------|-----------|-----------|--------|
| 3390 | 304 | 45 | 15 | 5 | 2 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 |
| 129667 | 207 | 28 | 2 | 54 | 2 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 |
| 188113 | 1201 | 178 | 2 | 27 | 1 | 5 | 1 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 2 |
| 180708 | 1001 | 135 | 14 | 54 | 2 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 |
| 158011 | 1301 | 217 | 1 | 69 | 2 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 |

5 rows × 36 columns

Dari kode program diatas, didapat output bahwa fitur yang tidak penting sudah di drop, seperti fitur

cmg,jkpst,dx2_d50_d89,dx2_f00_f99,dx2_l00_l99,proc52_57,proc24_27,dx2_j00_j99,dx2_r00_r99,proc14_23,proc68_70,proc58_62,diagprimer.

2. Scaling Dataset, dilakukan supaya setiap data yang ada pada dataset memiliki skala yang sama

```
x=data_new.drop(columns=['label'])
y = data_new['label'].values
```

```
#Mengubah skala data menjadi skala antara 0-1 dengan MinMaxScaler
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
X= scaler.fit_transform(X)
```

x

```
array([[0.40159681, 0.28273245, 0.91666667, ..., 0.          , 0.          ,
        0.          ],
       [0.47984032, 0.37760911, 0.08333333, ..., 0.          , 0.17391304,
        0.          ],
       [0.40439122, 0.32447818, 0.04166667, ..., 0.          , 0.          ,
        0.          ],
       ...,
       [0.03992016, 0.07020873, 0.91666667, ..., 0.          , 0.          ,
        0.          ],
       [0.36207585, 0.24098672, 0.04166667, ..., 0.          , 0.04347826,
        0.          ],
       [0.36526946, 0.22011385, 0.95833333, ..., 0.          , 0.          ,
        0.          ]])
```

3. Split Dataset

Split Dataset bertujuan untuk membagi data menjadi 2 bagian, yaitu Data Train, dan data Test. Dimana dataset Train 80% dan dataset Test 20%.

```
#Mengsplit data dengan menggunakan sklearn ( rasio 80:20)
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)
```


BAB 4

MODELING

Pada tahap *modeling* akan dijelaskan mengenai pemilihan teknik modelling, menghasilkan test design, membangun model, dan menilai model yang telah dibangun.. Data yang telah dipersiapkan dan juga telah dianalisis pada data *preparation* kemudian akan dibawa ke *modeling* setelah itu hasilnya akan menjelaskan mengenai masalah bisnis yang ditimbulkan selama proses *business understanding*. *Modeling* biasanya dilakukan dalam beberapa iterasi Pada penambahan data biasanya menjalankan beberapa model menggunakan parameter *default* dan setelah itu menyempurnakan parameter atau kembali ke data *preparation*.

4.1 Selection Modeling Technique

Teknik pemodelan yang digunakan pada proyek ini didorong oleh tujuan penambahan data yang ingin dicapai dalam proyek. Penerapan algoritma *support vector machine* cocok digunakan dalam teknik pemodelan dalam pengerjaan proyek ini dikarenakan SVM adalah algoritma pembelajaran terawasi yang sangat efektif digunakan untuk *classification*. Dalam algoritma SVM, pada data pelatihan, algoritma mencoba menemukan hyperplane optimal terbaik yang dapat digunakan untuk melakukan klasifikasi data. Biasanya dalam SVM akan bekerja dengan menemukan contoh yang paling mirip antar kelas sehingga akan dijadikan sebagai vektor pendukung.

Untuk menentukan model yang sesuai biasanya akan didasarkan pada pertimbangan berikut:

1. Tipe data yang tersedia untuk *mining*
2. Tujuan *data mining*
3. Persyaratan pemodelan khusus

4.1.1 Modeling Techniques

Teknik pemodelan yang digunakan pada proyek ini adalah algoritma *support vector machine* sesuai dengan tujuan *data mining* yaitu menggali *Discovering Knowledge* mengenai pola (*pattern*) item mengenai *Fraud Detection* menggunakan dataset BPJS Hackathon.

Algoritma *support vector machine* (SVM) adalah sebuah algoritma klasifikasi berdasarkan prinsip linear classifier yang mampu menyelesaikan permasalahan dengan waktu komputasi lebih cepat daripada SVM standar untuk data yang berukuran besar.

```
# import SVC classifier
from sklearn.svm import SVC
# import metrics to compute accuracy
from sklearn.metrics import accuracy_score
```

Untuk model pertama kita menggunakan kernel RBF

```
svmRBF = SVC(
    kernel = 'rbf',
    C=0.1,
    gamma = 1,
)
svmRBF.fit(X_train, y_train)
y_pred = svmRBF.predict(X_test)
print('Model accuracy kernel RBF : {0:0.4f}'.format(accuracy_score(y_test, y_pred)))
```

Model accuracy kernel RBF : 0.5648

```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0 | 0.56 | 0.61 | 0.58 | 20019 |
| 1 | 0.57 | 0.52 | 0.55 | 20025 |
| accuracy | | | 0.56 | 40044 |
| macro avg | 0.57 | 0.56 | 0.56 | 40044 |
| weighted avg | 0.57 | 0.56 | 0.56 | 40044 |

4.1.2 Modeling Assumptions

Dalam teknik pemodelan dengan *Support Vector Machine* memerlukan asumsi spesifik terhadap data, yaitu semua atribut memiliki distribusi yang sama, tidak ada missing value. Untuk atribut yang tidak Kategorikal (nominal) maka akan dilakukan pembuatan bin terlebih dahulu sebelum dilakukan penerapan algoritma SVM pada data tersebut.

4.2 Generate Test Design

Sebelum melakukan pembangunan model, perlu dilakukan perancangan terhadap bagaimana model akan diuji. Cara yang digunakan untuk menghasilkan test design yang *komprehensif* yaitu menentukan data yang akan menguji kriteria. Kriteria model yang dinilai bergantung pada data mining goals pada model yang akan dibangun. Tidak ada cara objektif untuk menilai model sampai disajikan pada algoritma secara langsung. Namun algoritma memerlukan aturan yang menghasilkan prediksi terhadap *Detection Fraud*.

4.2.1 Test Design

Desain pengujian (test design) merupakan gambaran langkah-langkah yang akan dilakukan untuk menguji model yang dihasilkan. Pada proyek ini, langkah-langkah untuk menguji model adalah sebagai berikut :

1. Mengekstrak test data yaitu record yang tidak digunakan dalam training set.
2. Menghitung instance yang benar di mana premisnya mengarah ke kesimpulan.
3. Menghitung confidence setiap aturan dari jumlah yang benar.
4. Mencetak aturan asosiasi terbaik dengan judul.

4.3 Build Model

Pada proses pembuatan model, terdapat tiga informasi yang akan digunakan dalam keputusan data *mining*, diantaranya:

1. *Parameter settings*

Parameter settings adalah pengaturan parameter yang mencakup catatan mengenai parameter yang memberikan hasil yang terbaik.

2. *Models*

Models dimana model aktual yang diproduksi

4.3.1 Parameter Settings

Pada sebagian besar teknik *modeling* mempunyai beberapa parameter yang dapat disesuaikan untuk mengamati dan mengendalikan proses *modeling*. Pada proyek ini menggunakan parameter C, kernel, dan gamma untuk menentukan nilai parameter-parameter model.

4.3.2 Models

Pada bagian ini, setelah menentukan parameter yang akan dipakai dan dibutuhkan pada proyek, langkah selanjutnya adalah mengeksekusi model untuk menghasilkan *result* atau *output* yang terlihat.

4.4 Assess Model

Assess model merupakan tahapan yang dilakukan untuk menilai kesesuaian model yang telah dibangun dengan kriteria sukses yang telah didefinisikan. Secara umum, hasil yang diperoleh dari pembangunan model dengan menggunakan algoritma SVM telah menghasilkan rule yang baik.

BAB 5

Evaluation

Pada tahap *Evaluation* (Evaluasi), akan dijelaskan mengenai evaluasi terhadap model untuk memprediksi Fraud Detection yang dihasilkan dengan menggunakan algoritma SVM. Evaluasi adalah fase interpretasi terhadap hasil *data mining*. Evaluasi dilakukan secara mendalam dengan tujuan agar hasil pada tahap *modelling* sesuai dengan sasaran yang ingin dicapai dalam tahap *business understanding*.

5.1 Evaluate Result

Tahap ini dilakukan untuk mengetahui performa SVM dengan menggunakan dataset yang diperoleh. Dari pemodelan yang dilakukan pada tahap sebelumnya, dilakukan implementasi menggunakan bahasa pemrograman python.

- SVM menggunakan kernel linear dengan nilai parameter C

```
#SVM menggunakan kernel linear dengan nilai parameter C
svmLinear = SVC(
    kernel = 'linear',
    C=1
)
svmLinear.fit(X_train, y_train)
y_pred = svmLinear.predict(X_test)
print('Model accuracy kernel Linear : {0:0.4f}'.format(accuracy_score(y_test, y_pred)))
```

Pada kode program diatas akan dilakukan evaluasi SVM menggunakan kernel linear dengan nilai parameter C dengan accuracy 0.5488

```
Model accuracy kernel Linear : 0.5448

from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

Berdasarkan kode program diatas didapatkan luaran seperti berikut :

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0 | 0.58 | 0.34 | 0.42 | 20019 |
| 1 | 0.53 | 0.75 | 0.62 | 20025 |
| accuracy | | | 0.54 | 40044 |
| macro avg | 0.55 | 0.54 | 0.52 | 40044 |
| weighted avg | 0.55 | 0.54 | 0.52 | 40044 |

- SVM menggunakan kernel Polynomial dengan nilai parameter C

```
#SVM menggunakan kernel Polynomial dengan nilai parameter C
svmPoly = SVC(
    kernel = 'poly',
    C=1,
    gamma = 0.01,
    degree =2
)
svmPoly.fit(X_train, y_train)
y_pred = svmPoly.predict(X_test)
print('Model accuracy kernel Polynomial : {0:0.4f}'.format(accuracy_score(y_test, y_pred)))
```

Pada kode program diatas akan dilakukan evaluasi SVM menggunakan Polynomial dengan nilai parameter C dengan accuracy 0.5459

```
Model accuracy kernel Polynomial : 0.5459

from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

Berdasarkan kode program diatas didapatkan luaran seperti berikut :

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0 | 0.54 | 0.66 | 0.59 | 20019 |
| 1 | 0.56 | 0.43 | 0.49 | 20025 |
| accuracy | | | 0.55 | 40044 |
| macro avg | 0.55 | 0.55 | 0.54 | 40044 |
| weighted avg | 0.55 | 0.55 | 0.54 | 40044 |

- SVM menggunakan kernel Sigmoid dengan nilai parameter C

```
#SVM menggunakan kernel Sigmoid dengan nilai parameter C
svmSigmoid = SVC(
    kernel = 'sigmoid',
    C=1,
    gamma = 0.1,
)
svmSigmoid.fit(X_train, y_train)
y_pred = svmSigmoid.predict(X_test)
print('Model accuracy kernel Sigmoid : {0:0.4f}'.format(accuracy_score(y_test, y_p
```

Pada kode program diatas akan dilakukan evaluasi SVM menggunakan Sigmoid dengan nilai parameter C dengan accuracy 0.5088

```
Model accuracy kernel Sigmoid : 0.5088
```

```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

Berdasarkan kode program diatas didapatkan luaran seperti berikut :

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0 | 0.51 | 0.50 | 0.50 | 20019 |
| 1 | 0.51 | 0.52 | 0.51 | 20025 |
| accuracy | | | 0.51 | 40044 |
| macro avg | 0.51 | 0.51 | 0.51 | 40044 |
| weighted avg | 0.51 | 0.51 | 0.51 | 40044 |

- SVM menggunakan kernel RBF dengan nilai parameter C

```
svmRBF = SVC(
    kernel = 'rbf',
    C=1,
    gamma = 1,
)
svmRBF.fit(X_train, y_train)
y_pred = svmRBF.predict(X_test)
print('Model accuracy kernel RBF : {0:0.4f}'.format(accuracy_score(y_test, y_p
```

Pada kode program diatas akan dilakukan evaluasi SVM menggunakan RBF dengan nilai parameter C dengan accuracy 0.5782

```
Model accuracy kernel RBF : 0.5782
```

```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

Berdasarkan kode program diatas didapatkan luaran seperti berikut :

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0 | 0.58 | 0.59 | 0.58 | 20019 |
| 1 | 0.58 | 0.57 | 0.57 | 20025 |
| accuracy | | | 0.58 | 40044 |
| macro avg | 0.58 | 0.58 | 0.58 | 40044 |
| weighted avg | 0.58 | 0.58 | 0.58 | 40044 |

Pada modeling awal akurasi model yang didapatkan adalah 0.5448. Untuk meningkatkan akurasi dilakukan parameter tuning, yang dimana parameter yang di tuning meliputi C, kernel, dan gamma. Setelah melakukan parameter tuning, didapatkan kenaikan akurasi model menjadi 0.5782 dengan rincian Accuracy 58%, Precision 58%, dan recall 59%.

5.2 Evaluate Process

Tahap ini memeriksa kembali tahapan dari awal untuk memastikan bahwa tidak ada faktor penting dalam proses tersebut yang terabaikan atau terlewat. Berdasarkan hasil peninjauan proses awal proyek data mining dengan metodologi SVM maka dapat dipahami bahwa:

1. Proses eksplorasi data akan membantu dalam memilih atribut yang berkaitan dengan *Fraud Detection*.
2. Data Preparation, khususnya pada proses *data cleaning* dan *transforming data*, sehingga data yang diperoleh dapat menghasilkan model yang baik.
3. Sangat penting untuk tetap fokus pada masalah bisnis yang dihadapi, karena setelah data siap dianalisis, maka akan dilakukan tahap pemodelan. Business understanding sangat penting dalam memutuskan bagaimana menerapkan hasil yang diperlukan dalam *Fraud Detection*

5.3 Determine Next Steps

Tahapan ini menentukan langkah apa yang akan diambil selanjutnya. Berdasarkan hasil evaluasi terhadap model yang digunakan dengan algoritma SVM, jika telah menghasilkan yang terbaik maka diputuskan pengerjaan proyek akan dilanjutkan ke tahap akhir yakni deployment.

BAB 6

DEPLOYMENT

Tahap keenam yaitu untuk melakukan prediksi *Fraud Detection Train* adalah deployment. Pada bab ini akan dijelaskan mengenai perencanaan fase penyebaran atau penggunaan model yang sudah dihasilkan, perencanaan pemantauan dan pemeliharaan.

6.1 Plan Deployment

Pada fase plan deployment ini, model yang telah terbentuk pada fase modelling akan digunakan sesuai dengan tujuan data mining yang dibutuhkan. Penggunaan model yang telah dihasilkan akan memerlukan dataset yang sesuai dengan tujuan penggunaannya. Pada kasus proyek ini algoritma SVM akan digunakan sesuai data *Fraud Detection Train* yang sudah diperbaharui secara real time. Data yang sudah diperbaharui tersebut akan digunakan untuk memprediksi keakuratan terhadap *fraud* yang terjadi menggunakan model yang sudah dirancang. Namun, jika dataset yang akan digunakan masih kotor atau terdapat record yang tidak memiliki nilai (missing value) serta terdapat beberapa variabel yang tidak dibutuhkan untuk memprediksi *Fraud Detection Train*, maka dataset tersebut harus dibersihkan terlebih dahulu (data preprocessing) sesuai penjelasan pada bab 3. Sehingga proses pemodelan nantinya akan berjalan dengan baik dengan spesifik atribut atau parameter yang memiliki distribusi yang sesuai. Selanjutnya dataset tersebut akan diproses sesuai dengan jenis tipe datanya dan akan diproses menggunakan model yang telah dihasilkan. Dari penggunaan model tersebut, maka akan dihasilkan beberapa rule sesuai dengan kebutuhan objek yang dibutuhkan.

6.2 Plan Monitoring and Maintenance

Dalam monitoring dan maintenance adalah untuk menentukan apakah prediksi yang digunakan dengan algoritma SVM sudah efektif. Apakah atribut yang digunakan tepat sehingga memenuhi parameter yang telah ditentukan. Dikarenakan proyek yang dilakukan di masa depan dapat menghasilkan model yang lebih kompleks, maka monitoring akan ditingkatkan. Alternatif yang memungkinkan adalah dengan mencoba pembuatan model untuk prediksi dengan tepat dan akurat yang sangat dibutuhkan dalam pengerjaan proyek data mining.

6.3 Produce Final Report

Pada akhir proyek, tim proyek membuat laporan akhir dari penambangan data yang telah dilakukan. Report tersebut mencakup ringkasan dari proyek yang dilakukan, deliverables yang dihasilkan dari proyek, dan mengorganisir hasil yang diperoleh untuk disampaikan kepada audience. Dalam proyek ini, final report yang dimaksud mencakup dokumen pengerjaan proyek, file presentasi mencakup langkah-langkah pengerjaan yang dilakukan, poster, dan video presentasi untuk menyampaikan tahapan dan hasil yang diperoleh.

6.4 Review Project

Review project digunakan untuk menilai baik, buruknya proyek yang telah dibangun, apa yang telah selesai dan yang perlu dilakukan perbaikan kedepannya. Dalam pengerjaan proyek ini, tim proyek terlibat dalam pengerjaan proyek dari awal hingga akhir sehingga mampu mendapatkan pemahaman lebih detail mengenai eksplorasi data pada dataset yang digunakan, tahapan pemrosesan data untuk mendapatkan data yang siap digunakan pada penerapan algoritma SVM. Selain itu, tim proyek juga mendapatkan pemahaman dengan menerapkan secara langsung bagaimana penerapan algoritma dalam melakukan data mining task.

DAFTAR PUSTAKA

- [1] Q. A. Al-Radaideh and E. Al Nagi, "Using Data Mining Techniques to Build a Classification Model for Predicting Employees Performance," *Int. J. Adv. Comput. Sci. Appl.*, vol. 3, no. 2, 2012
- [2] N. Ketui, W. Wisomka, and K. Homjun, "Association Rule Mining with Permutation for Estimating Students Performance and Its Smart Education System," *J. Comput.*, vol. 30, no. 2, pp. 93–102, 2019.
- [3] M. Jozsef, R. Szabolcs, "Support vector machine and fuzzy logic", "Acta Polytechnica Hungarica Vol.13, No. 5.207-210, 2016.
- [4] C. Ivo Rally Drajana, "Metode support vector machine dan forward selection prediksi pembayaran pembelian bahan baku kopra, "ILKOM Jurnal Ilmiah Volume 9 Nomor 2. 116-117, Agustus 2017
- [5] B. Saeed, A. Akbarzadeh, M. Zarrabi, "using pca combined svm in the classification of eutrophication in dez reservoir (iran)", "Environmental Engineering and Management Journal Vol. 16, No. 9,. 2140-2141, September 2017