DOKUMEN PROYEK
12S4054 - PENAMBANGAN DATA
Fraud Detection (Binary Classification) BPJS Hackathon Using SVM Algorithm

Disusun Oleh:

12S19051 Corri Hutahaean

12S19052 Mulyani Gabe Sayoni Simanjuntak

12S19053 Elysa Noelia Pangaribuan

PROGRAM STUDI SARJANA SISTEM INFORMASI
FAKULTAS INFORMATIKA DAN TEKNIK ELEKTRO
INSTITUT TEKNOLOGI DEL
TAHUN 2022/2023

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB 1 BUSINESS UNDERSTANDING

Business Understanding adalah langkah pertama dalam CRISP-DM yang secara general digunakan untuk mendefinisikan proyek, tujuan dan kebutuhan dari sudut pandang bisnis, yang kemudian akan menerjemahkan pengetahuan yang sudah diperoleh ke dalam pendefinisian masalah pada data mining sehingga dapat dilakukan penyesuaian terhadap tujuan bisnis sehingga model terbaik dapat dibangun. Tahap *business understanding* juga merupakan tahap yang digunakan untuk mengetahui dan menentukan rencana dan strategi untuk mencapai tujuan yang sudah didefinisikan di awal. Pada tahap ini diperlukan pengetahuan dari objek bisnis tertentu, yaitu bagaimana membangun atau mendapatkan data, dan bagaimana untuk mencocokkan tujuan pemodelan untuk tujuan bisnis sehingga model terbaik dapat dibangun [1].

Pada bab ini akan menjelaskan pemahaman mengenai aktivitas dalam menentukan sasaran bisnis, memahami situasi bisnis, menerjemahkan tujuan atau sasaran bisnis ke dalam data mining.Dalam sistem yang akan dikembangkan oleh penulis meliputi 4 tahap dalam business understanding, yaitu: *determine business objectives, assess situation, determine data mining goals,* dan *produce project plan* [2].

1. 1 Determine Business Objective

Menganalisis data agar benar-benar memahami dari perspektif bisnis, mengenai apa yang ingin dicapai. Tujuan dilakukannya analisis adalah untuk mendapatkan faktor-faktor penting yang dapat mempengaruhi hasil proyek sehingga penelitian tidak akan menghasilkan jawaban yang benar atas pernyataan yang salah.

BPJS Kesehatan beroperasi sejak 1 Januari 2014 sampai sekarang mengalami banyak tantangan dalam melaksanakan program Jaminan Kesehatan Nasional (JKN), yaitu salah satunya mencegah terjadinya tindak kecurangan (fraud). Fraud adalah tindakan yang dilakukan untuk mencari keuntungan dengan melakukan kesalahan terhadap kebenaran yang ada. Tindak kecurangan dalam BPJS berpotensi dilakukan oleh peserta seperti membuat pernyataan yang tidak benar dalam hal eligibilitas (memalsukan status kepesertaan) untuk memperoleh pelayanan kesehatan, memanfaatkan halnya untuk pelayanan yang tidak perlu dengan cara memalsukan kondisi kesehatan, memberikan gratifikasi kepada pemberi pelayanan agar bersedia memberi pelayanan yang tidak sesuai, memanipulasi penghasilan agar tidak perlu membayar iuran terlalu besar[1]

1.2 DeterSituation Assessment

Pada tahapan ini akan dilakukan pencarian fakta yang lebih terperinci mengenai semua sumber daya, kendala, asumsi, dan faktor lainnya yang harus dipertimbangkan dalam menentukan tujuan analisis data dan rencana penelitian. Tahapan ini bertujuan untuk

[1] https://bpjs-kesehatan.go.id/bpjs/

memperluas detail dari analisis yang dihasilkan pada tahapan pertama. Sumber daya yang akan digunakan dalam penelitian adalah data dari BPJS Hackaton yang terdiri dari 200217 observasi dan 53 variable. Pada zaman sekarang sangat dibutuhkan sistem *fraud detection* yaitu untuk melakukan pendeteksian atau melakukan investigasi suatu kegiatan yang mencurigakan dan sekaligus mencegah terjadinya kecurangan. Tujuan pengerjaan proyek ini adalah membangun sebuah model dengan penggunaan teknik dalam *data mining* untuk mengetahui *fraud* dari data BPJS Hackaton dengan hasil apakah data *fraud* atau *non-fraud.*

1.3 Determine Data Mining Goal

Pada tahap ini, akan ditentukan tujuan dalam terminologi bisnis. Tujuan data mining menyatakan tujuan proyek ini. Dan tujuan dari proyek ini adalah untuk mengembangkan sebuah model data mining untuk melakukan prediksi potensi terjadinya *fraud* pada klaim pelayanan Rumah Sakit berdasarkan *dataset train* yang terdiri dari 200.217 observasi dan 53 variabel.

1.4 Produce Project Plan

Pada tahap ini, rencana yang akan dilaksanakan untuk mencapai tujuan data dijelaskan untuk pertambangan dan mencapai tujuan komersial. Rencana yang dibuat harus dapat ditentukan langkah-langkah yang harus diambil selama sisa proyek, termasuk pemilihan alat dan teknik awal. Algoritma yang akan digunakan dalam penelitian ini adalah Algoritma *SVM* yang akan membantu melakukan klasifikasi terhadap kecurangan pelayanan rumah sakit.

BAB 2 DATA UNDERSTANDING

Data Understanding adalah tahap pengumpulan data yang akan dilanjutkan dengan sebuah proses untuk memperoleh pemahaman mendalam mengenai data, mengidentifikasi kualitas data. serta memungkinkan untuk melakukan deteksi apabila terdapat sebuah bagian unik dari data yang baik digunakan sebagai hipotesis terhadap informasi yang tersembunyi. Tahap *data understanding* memberikan fondasi analitik dengan membuat ringkasan dan melakukan identifikasi potensi masalah dalam data yang harus dilakukan secara cermat.

2.1 Collect Initial Data

Pada tahapan ini akan dilakukan pengumpulan data yang telah diberikan dosen pengampu yang diambil dari *case* salah satu lomba yaitu Hackaton. Dataset tersebut memiliki format *file* CSV (*Comma Separated Values*) sehingga datanya bersifat statis dan terstruktur. Ukuran data yang digunakan dalam kasus *fraud detection* menggunakan *Super Vector Machine adalah* 10611501.

```
In [1]: #import library
        import matplotlib.pyplot as plt
        import numpy as np
        import pandas as pd

In [2]: #Load dataset
        df = pd.read_csv("D:/PERKULIAHAN/Semester 7/Penambangan data/Week 1/Materi/fraud_detection_train.csv")

In [3]: df.sample(5)
```

Out[3]:

	visit_id	kdkc	dati2	typeppk	jkpst	umur	jnspelsep	los	cmg	severitylevel	...	proc63_67	proc68_70	proc71_73	proc74_75	proc76_77	proc78_79
15882	15883	2101	309	KM	P	52	2	0	H	0	...	0	0	0	0	0	0
140020	140021	1311	192	C	P	44	2	0	I	0	...	0	0	0	0	0	0
137538	137539	1601	304	SC	P	22	2	0	Q	0	...	0	0	0	0	0	0
47862	47863	1007	129	B	P	29	1	1	K	1	...	0	0	0	0	0	0
64971	64972	1004	224	B	L	0	2	0	F	0	...	0	0	0	0	0	0

5 rows × 53 columns

Gambar 1 Informasi ukuran data

2.2 Describe Data

Pada tahapan ini akan dilakukan pendeskripsian terhadap data yang akan digunakan berupa keterangan mengenai format data, jumlah data, jumlah atribut dan fitur yang digunakan pada pengerjaan proyek. Dataset yang digunakan untuk melakukan prediksi terjadinya *fraud* pada klaim pelayanan Rumah Sakit. Untuk membaca dataset yang akan digunakan terlebih dahulu import library pandas untuk membaca data.

```
In [4]: df.size
Out[4]: 10611501
```

Gambar 2. Kode untuk membaca dataset.

Data yang digunakan berjumlah 200.217 data dan terdapat 53 variabel, terdapat 4 atribut bertipe data kategorikal dan 49 atribut bertipe data numerik. Pada tahapan deskripsi data ini juga dapat memberikan informasi apa saja yang dapat digunakan untuk melakukan

implementasi pada sistem yang dibangun. Untuk melihat dimensi dataset digunakan fungsi df.shape. Fungsi df.columns pada pandas digunakan untuk melihat kolom yang ada pada dataset.

```
In [3]: df.shape

Out[3]: (200217, 53)
```

Gambar 3. Kode untuk melihat bentuk data

Gambar 3 Kode untuk menampilkan total data dan variabel

```
In [11]: df.columns

Out[11]: Index(['visit_id', 'kdkc', 'dati2', 'typeppk', 'jkpst', 'umur', 'jnspelsep',
       'los', 'cmg', 'severitylevel', 'diagprimer', 'dx2_a00_b99',
       'dx2_c00_d48', 'dx2_d50_d89', 'dx2_e00_e90', 'dx2_f00_f99',
       'dx2_g00_g99', 'dx2_h00_h59', 'dx2_h60_h95', 'dx2_i00_i99',
       'dx2_j00_j99', 'dx2_koo_k93', 'dx2_l00_l99', 'dx2_m00_m99',
       'dx2_n00_n99', 'dx2_o00_o99', 'dx2_p00_p96', 'dx2_q00_q99',
       'dx2_r00_r99', 'dx2_s00_t98', 'dx2_u00_u99', 'dx2_v01_y98',
       'dx2_z00_z99', 'proc00_13', 'proc14_23', 'proc24_27', 'proc28_28',
       'proc29_31', 'proc_32_38', 'proc39_45', 'proc46_51', 'proc52_57',
       'proc58_62', 'proc63_67', 'proc68_70', 'proc71_73', 'proc74_75',
       'proc76_77', 'proc78_79', 'proc80_99', 'proce00_e99', 'procv00_v89',
       'label'],
      dtype='object')
```

Gambar 4. Kode untuk melihat kolom pada dataset

Pada hasil yang telah diberikan nama dari 53 atribut yang ada pada dataset. Selanjutnya, untuk melihat detail statistik seperti persentil, rata-rata, standar deviasi dan lainnya dari atribut dalam dataset digunakan fungsi df.describe().

```
In [7]: df.describe()
```

Out[7]:

	visit_id	kdkc	dati2	umur	jnspelsep	los	severitylevel	dx2_a00_b99
count	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
std	57797.813761	574.486224	107.226676	23.095928	0.470294	5.639751	0.725227	0.162484
min	1.000000	101.000000	1.000000	0.000000	1.000000	0.000000	0.000000	0.000000
25%	50055.000000	903.000000	114.000000	18.000000	1.000000	0.000000	0.000000	0.000000
50%	100109.000000	1101.000000	169.000000	39.000000	2.000000	0.000000	0.000000	0.000000
75%	150163.000000	1314.000000	232.000000	56.000000	2.000000	2.000000	1.000000	0.000000
max	200217.000000	2606.000000	528.000000	109.000000	2.000000	592.000000	3.000000	4.000000

8 rows × 49 columns

Gambar 5 Kode untuk detail statistik dataset

Berikut adalah atribut yang terdapat pada dataset fraud_detection_tain.

Tabel 1 Keterangan Atribut 1

No	Variabel	Tipe Variabel	Deskripsi
1	visit_id	int64	id kunjungan
2	Kdkc	int64	kode wilayah kantor cabang BPJS Kesehatan
3	dati2	int64	kode kabupaten/kota
4	typeppk	object	kode tipe Rumah Sakit
5	jkpst	object	jenis kelamin peserta JKN-KIS
6	umur	int64	umur peserta saat mendapatkan pelayanan rumah sakit
7	jnspelsep	int64	tingkat pelayanan; 1:rawat inap; 2: rawat jalan
8	los	int64	lama peserta dirawat di rumah sakit
9	cmg	object	klasifikasi CMG (Case Mix Group)

10	severitylevel	int64	tingkat urgensi
11	diagprimer	object	diagnosa primer
12	dx2_..._...	int64	diagnosa sekunder
13	proc.._...	int64	kode kelompok procedure
14	label	int64	flag fraud; 1:fraud; 0:tidak fraud

2.3. Explore Data

Pada bagian ini, dataset ditelaah untuk mendapatkan informasi terkait kondisi dari dataset. Penelaahan terhadap data dimulai dengan memperhatikan informasi terkait kolom yang terdapat di dalam dataset beserta data deskripsinya. Eksplorasi data dimulai dengan memperhatikan informasi terkait tipe.

Pembagian Data

a. Pembagian data fraud dan non fraud berdasarkan Label

```
In [11]: #Membagi data menjadi menjadi 2 berdasarkan label (fraud/non-fraud)
         group = df.groupby("label")

In [12]: df_fraud = group.get_group(1)
         df_non_fraud = group.get_group(0)

In [13]: df_fraud.sample(5)
```

Out[13]:

	visit_id	kdkc	dati2	typeppk	jkpst	umur	jnspelsep	los	cmg	severitylevel	...	proc63_67	proc68_70	proc71_73	proc74_75	proc76_77	proc78_79
17838	17839	1002	133	B	L	54	1	9	L	3	...	0	0	0	0	0	0
25487	25488	1107	150	SD	L	18	2	0	Q	0	...	0	0	0	0	0	0
52616	52617	802	103	D	P	55	1	2	L	1	...	0	0	0	0	0	0
57166	57167	1101	173	SC	P	6	2	0	Q	0	...	0	0	0	0	0	0
11539	11540	2301	240	B	L	11	2	0	Q	0	...	0	0	0	0	0	0

5 rows × 53 columns

```
In [15]: df_non_fraud.sample(5)
```

Out[15]:

	visit_id	kdkc	dati2	typeppk	jkpst	umur	jnspelsep	los	cmg	severitylevel	...	proc63_67	proc68_70	proc71_73	proc74_75	proc76_77	proc78_79
117906	117907	1104	142	B	L	36	2	0	Q	0	...	0	0	0	0	0	0
162545	162546	501	82	KI	P	0	1	8	P	2	...	0	0	0	0	0	0
150136	150137	905	116	SC	L	21	1	4	K	1	...	0	0	0	0	0	0
175583	175584	1315	205	SD	P	28	2	0	Q	0	...	0	0	0	0	0	0
119466	119467	1603	299	C	L	41	2	0	Q	0	...	0	0	0	0	0	0

5 rows × 53 columns

Gambar 6 kode untuk membagi data

b. Pasien yang melakukan kecurangan berdasarkan gender.

```
In [16]: #melihat rata2 umur pasien yang melakukan kecurangan berdasarkan gender
         df_fraud.groupby('jkpst', as_index=False).umur.mean()
```

Out[16]:

	jkpst	umur
0	L	36.330913
1	P	37.893959

Gambar 7 untuk menampilkan data fraud berdasarkan gender

c. Deskripsi data pasien yang melakukan kecurangan mengenai umur.

```
In [17]: #melihat deskripsi data umur pasien yang curang
         df['umur'].describe()
```

```
Out[17]: 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
```

```
In [18]: import matplotlib.pyplot as plt

         plt.style.use('ggplot')
         ages = df['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.
```

Gambar 8 tampilan data fraud mengenai umur

d. Info data setiap fitur pada dataset

In [19]:
```
#check info dataset
df.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_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
 25  dx2_o00_o99     200217 non-null  int64
 26  dx2_p00_p96     200217 non-null  int64
```

Gambar 9 tampilan fitur pada dataset

```
dtypes: int64(49), object(4)
memory usage: 81.0+ MB
```

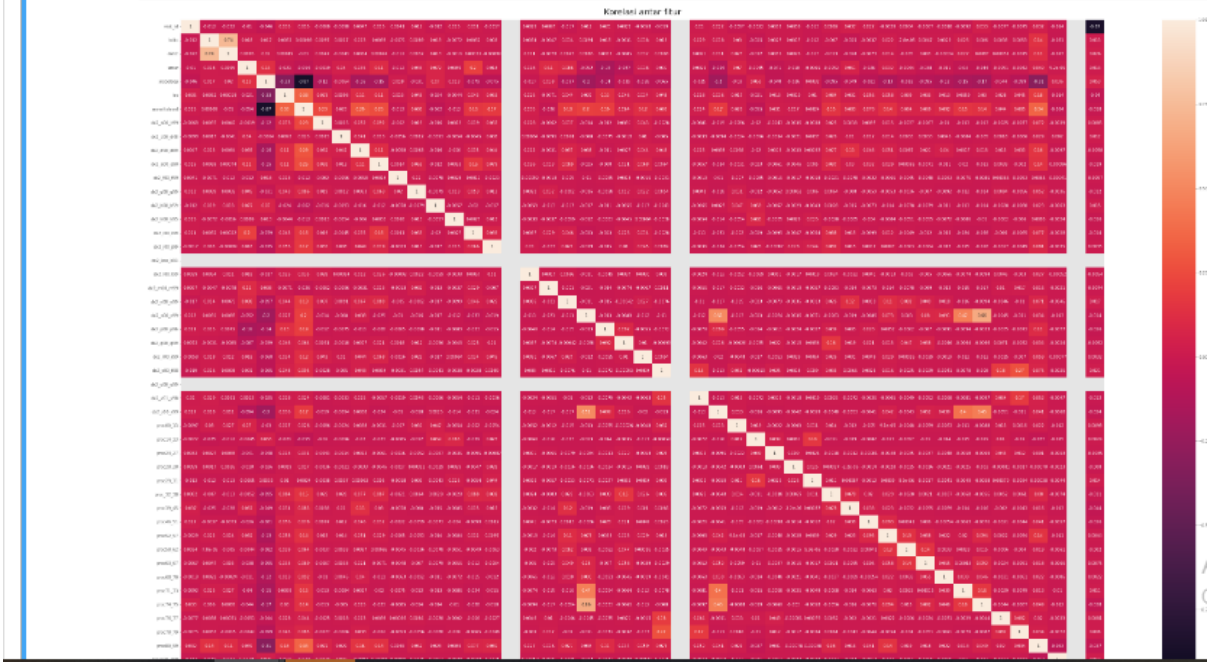
Gambar 10. Kode untuk melihat type atribut

Pada *dataset* tidak terdapat *missing value,* dimana tidak ada yang bernilai '*null'* dan 'NaN'

e. Korelasi antar Fitur

```
In [22]: import seaborn as sns
         import matplotlib.pyplot as plt

In [23]: #Visualisasi korelasi antara fitur
         plt.figure(figsize=(50, 30))
         heatmap = sns.heatmap(df.corr(), vmin=-1, vmax=1, annot=True)
         heatmap.set_title('Korelasi antar fitur', fontdict={'fontsize':20}, pad=12);
```



Gambar 11 Korelasi Fitur sebelum di drop 1

Fitur yang tidak memiliki korelasi dengan fitur lainnya adalah fitur *'procv00_v89','dx2_u00_u99','dx2_koo_k93',* sehingga akan menggunakan fungsi *drop* untuk menghapusnya.

Setelah berhasil mendapatkan korelasi antar atribut, maka langkah selanjutnya melakukan penelaahan terhadap value dari masing-masing atribut. Dari tampilan visualisasi histogram dibawah. Tampilan value atribut dalam dataset, maka didapatkan hal-hal berikut ini :

1. Atribut yang memiliki variasi value terbanyak adalah atribut kdkc, dati2, dan umur.
2. Atribut kdkc dengan value 1000 memiliki frekuensi tertinggi dan value 2250 memiliki frekuensi terendah.
3. Atribut kdkc menunjukkan kode wilayah kantor cabang BPJS Kesehatan, yang menunjukkan bahwa kode kdkc dengan value sekitar 1000 memiliki jumlah pasien terbanyak.
4. Atribut dati2 dengan value 100 memiliki frekuensi tertinggi dan value 500 terendah yang menunjukkan bahwa kabupaten dengan kode 100 memiliki jumlah pasien tertinggi.
5. Untuk umur pasien dengan jumlah terbanyak adalah pasien dengan umur sekitar 0

bulan dan umum pasien dengan jumlah terkecil adalah umur 80.

BAB 3 DATA PREPARATION

Data preparation merupakan langkah setelah dilakukannya pengumpulan data awal yang telah dilakukan pada fase crisp-dm 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*. Pada bab ini akan dijelaskan mengenai proses apa saja yang akan dilakukan untuk mempersiapkan data seperti *sorting, cleaning, construction, binning dan normalization.*

3.1 Cleaning Data

Tahapan yang dilakukan saat data yang akan diproses masih data kotor yang memiliki missing value yang akan mempersulit proses data.

3.1.1. Fungsi .dropna ()

Fungsi *drop* akan menghapus fitur yang tidak penting dan tidak memiliki keterkaitan antar fitur lainnya.

```
In [52]: df.drop(['visit_id', 'procv00_v89', 'dx2_koo_k93', 'dx2_u00_u99', 'proce00_e99',
```

```
In [34]: df.sample(10)
```

Out[34]:

	kdkc	typeppk	jkpst	umur	jnspelsep	los	cmg	severitylevel	diagprimer	dx2_a00_b99
48750	2605	D	L	55	1	4	S	1	s00_t98	0
60780	905	C	P	2	2	0	Q	0	q00_q99	0
17496	1007	D	P	38	1	3	U	1	k00_k93	0
115593	1808	SC	P	26	2	0	U	0	h60_h95	0
183026	1003	SC	L	28	2	0	Q	0	h60_h95	0
58714	902	SC	P	0	1	5	A	1	r00_r99	0
122233	1005	B	P	68	2	0	Q	0	z00_z99	0
114776	1314	SB	L	63	1	4	M	1	s00_t98	0
59764	1308	A	L	16	1	2	D	1	d50_d89	0
132118	2201	A	P	24	1	1	O	2	o00_o99	0

10 rows × 47 columns

Gambar 12 Fungsi Drop

Setelah melakukan fungsi *drop* maka jumlah fitur berkurang, yaitu dari 53 fitur menjadi 47 fitur.

3.2 Data Transformation

```
In [18]: #Untuk mentransformasi data kategorikal tersebut kita menggunakan Laber Encoder
         from sklearn import preprocessing
         lab_enc = preprocessing.LabelEncoder()
```

Gambar 13 Kode untuk mentransformasi data

```
In [19]: #Mentransformasi setiap data kategorika
         df['typeppk'] = lab_enc.fit_transform(df[['typeppk']])
         df['jkpst'] = lab_enc.fit_transform(df[['jkpst']])
         df['cmg'] = lab_enc.fit_transform(df[['cmg']])
         df['diagprimer'] = lab_enc.fit_transform(df[['diagprimer']])

C:\Users\User\anaconda3\lib\site-packages\sklearn\preprocessing\_label.py:115:
DataConversionWarning: A column-vector y was passed when a 1d array was expecte
d. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

Gambar 14 Kode untuk mentransformasi data kategorikat

```
In [55]: df.sample(5)
```

Out[55]:

	kdkc	typeppk	jkpst	umur	jnspelsep	los	cmg	severitylevel	diagprimer	dx2_a00_b99
175484	902	23	1	0	2	0	15	0	15	0
127232	2201	23	1	46	2	0	16	0	4	0
100282	604	2	1	35	2	0	16	0	6	0
140718	201	1	0	39	2	0	13	0	20	0
20639	1110	24	0	33	2	0	8	0	16	0

5 rows × 47 columns

Gambar 15 kode untuk menampilkan 5 sampel Data

3.3 Feature Selection

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

X = df.drop(columns=['label'])
y = df['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       visit_id  5.010820e+09
1           kdkc  9.795663e+03
2          dati2  8.669032e+03
7            los  7.942436e+03
3        typeppk  5.990419e+03
49     proc80_99  5.607761e+02
5            umur  5.280462e+02
39     proc39_45  3.958280e+02
46     proc74_75  3.322112e+02
26   dx2_p00_p96  3.239591e+02
40     proc46_51  1.556103e+02
29   dx2_s00_t98  1.388402e+02
32   dx2_z00_z99  1.118375e+02
14   dx2_e00_e90  8.943101e+01
```

```
9   severitylevel  7.953913e+01
6       jnspelsep  7.510131e+01
25    dx2_o00_o99  7.336312e+01
24    dx2_n00_n99  6.589106e+01
17    dx2_h00_h59  6.161282e+01
18    dx2_h60_h95  5.611181e+01
50    proce00_e99  5.416570e+01
19    dx2_i00_i99  4.456577e+01
37      proc29_31  3.921469e+01
45      proc71_73  3.447129e+01
16    dx2_g00_g99  3.368169e+01
38     proc_32_38  3.317639e+01
12    dx2_c00_d48  2.939354e+01
31    dx2_v01_y98  2.694772e+01
27    dx2_q00_q99  2.599315e+01
33      proc00_13  2.548347e+01
48      proc78_79  1.732188e+01
47      proc76_77  1.684259e+01
11    dx2_a00_b99  1.536247e+01
36      proc28_28  1.300434e+01
43      proc63_67  1.230153e+01
23    dx2_m00_m99  1.226057e+01
8             cmg  9.608012e+00
4           jkpst  8.074668e+00
13    dx2_d50_d89  7.059972e+00
15    dx2_f00_f99  6.720079e+00
22    dx2_l00_l99  6.279142e+00
41      proc52_57  4.619877e+00
35      proc24_27  2.765082e+00
20    dx2_j00_j99  2.616805e+00
28    dx2_r00_r99  2.183154e+00
34      proc14_23  1.275097e+00
44      proc68_70  1.028570e+00
42      proc58_62  8.847172e-01
10     diagprimer  6.889055e-01
```

.

Gambar 16 kode untuk menyeleksi data.

Fitur *selection* yang akan digunakan dalam pengerjaan proyek ini adalah menggunakan fitur *Select KBes* yang akan melakukan perankingan fitur berdasarkan pemilihan K-Best. Berdasarkan pe-renkingan fitur maka yang memiliki satu digit *score* akan di *drop*, karena pada pemodelan tidak akan berdampak.

a. *Drop* Fitur

Fungsi *Drop* pada perintah di bawah ini adalah untuk menghilangkan atau menghapus beberapa fitur yang tidak penting dan tidak memiliki pengaruh apapun pada proses preprocessing. Pada fitur ini atribut cmg, jkpst, dx2_d50, dx2_f00_f99 dan lainnya akan dihapus karena memiliki value yang tidak dibutuhkan saat proses modeling dilakukan.

```
#Drop fitur yang tidak penting
df_new=df.drop(['cmg','jkpst','dx2_d50_d89','dx2_f00_f99','dx2_l00_l99','proc52_
```

```
df.sample(5)
```

	kdkc	typeppk	jkpst	umur	jnspelsep	los	cmg	severitylevel	diagprimer	dx2_a00_b99
80462	405	2	1	63	1	2	7	2	6	0
193676	1003	23	0	24	2	0	5	0	4	0
40350	102	2	0	48	1	6	9	1	9	0
9532	903	23	0	0	1	3	15	1	15	0
156930	1018	1	0	8	2	0	16	0	2	0

5 rows × 47 columns

Gambar 17 Drop Fitur use K-Best

b. Info Data atribut

```
In [23]: #check info dataset
         df.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  int32
 4   jkpst           200217 non-null  int32
 5   umur            200217 non-null  int64
 6   jnspelsep       200217 non-null  int64
 7   los             200217 non-null  int64
 8   cmg             200217 non-null  int32
 9   severitylevel   200217 non-null  int64
 10  diagprimer      200217 non-null  int32
 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
 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
```

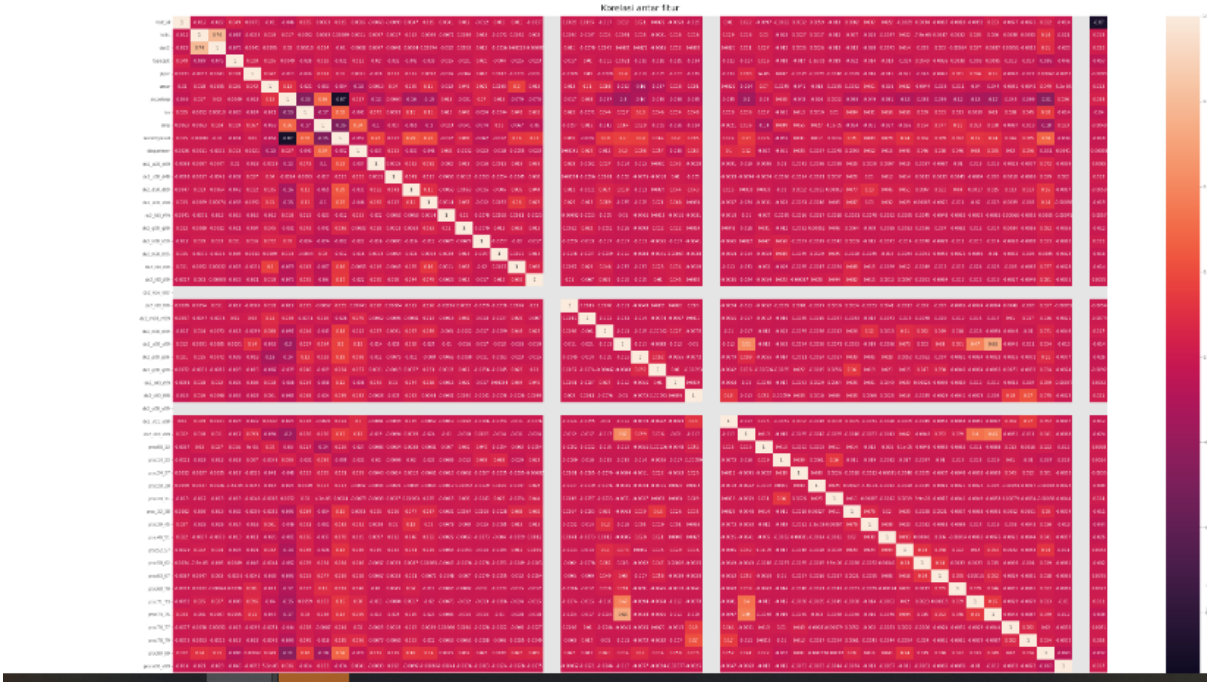
```
dtypes: int32(4), int64(49)
memory usage: 77.9 MB
```

Gambar 18 Kode untuk menampilkan info data Preparation

c. Korelasi antar Fitur setelah Data Preparation

```
In [37]: import seaborn as sns
         import matplotlib.pyplot as plt

In [38]: #Visualisasi korelasi antara fitur
         plt.figure(figsize=(50, 30))
         heatmap = sns.heatmap(df.corr(), vmin=-1, vmax=1, annot=True)
         heatmap.set_title('Korelasi antar fitur', fontdict={'fontsize':20}, pad=12);
```



Gambar 19 Kode untuk menampilkan korelasi antar atribut

3.4. Scaling Dataset

Scaling Dataset bertujuan untuk menyamakan skala pada setiap data yang ada pada *dataset.*

```
In [39]: X=df.drop(columns=['label'])
         y = df_new['label'].values

In [40]: #Mengubah skala data menjadi skala antara 0-1 dengan MinMaxScaler
         from sklearn.preprocessing import MinMaxScaler
         scaler = MinMaxScaler()
         X= scaler.fit_transform(X)

In [41]: X

Out[41]: array([[0.00000000e+00, 4.01596806e-01, 2.82732448e-01, ...,
                 0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
                [4.99460583e-06, 4.79840319e-01, 3.77609108e-01, ...,
                 1.73913043e-01, 0.00000000e+00, 0.00000000e+00],
                [9.98921165e-06, 4.04391218e-01, 3.24478178e-01, ...,
                 0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
                ...,
                [9.99990011e-01, 3.99201597e-02, 7.02087287e-02, ...,
                 0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
                [9.99995005e-01, 3.62075848e-01, 2.40986717e-01, ...,
                 4.34782609e-02, 0.00000000e+00, 0.00000000e+00],
                [1.00000000e+00, 3.65269461e-01, 2.20113852e-01, ...,
                 0.00000000e+00, 0.00000000e+00, 0.00000000e+00]])
```

Gambar 20 Scaling Dataset

3.5. Split Data

Kemudian lakukan *split Data*

```
In [42]: #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)
```

Gambar 21 Split Data di scaling

BAB 4 MODELLING

Pada bab ini akan menjelaskan mengenai pemilihan teknik modelling, menghasilkan test design, membangun model, dan menilai model yang telah dibangun. Pada modelling ini, data preparation akan dioperasikan yang kemudian akan menjelaskan masalah bisnis yang ditimbulkan selama proses business understanding.

4.1 Selection Modelling Technique

Teknik pemodelan yang digunakan pada proyek ini didorong oleh tujuan penambangan 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 Modelling Techniques

Teknik pemodelan yang digunakan pada proyek ini adalah algoritma support vector machine (SVM) 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.

MODELING

In [43]:
```
# 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

In [45]:
```
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 : 0.9944
```

In [46]:
```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.99      0.99      0.99     20019
           1       0.99      0.99      0.99     20025

    accuracy                           0.99     40044
   macro avg       0.99      0.99      0.99     40044
weighted avg       0.99      0.99      0.99     40044
```

Gambar 22 Modelling

4.1.2 Modelling Assumptions

Pada teknik ini akan dilakukan teknik pemodelan dengan *Support Vector Machine (SVM)* yang memerlukan asumsi spesifik terhadap data, yaitu semua atribut yang sama, tidak ada *missing value*. Pembuatan bin terlebih dahulu sebelum dilakukan penerapan algoritma SVM pada data tersebut untuk atribut yang tidak kategorikal (nominal).

4.2. Generate Test Design

Sebelum dilakukannya pembangunan model, perlu dilakukan perancangan terhadap bagaimana model akan diuji. Untuk mendapatkan hasil test design yang komprehensif yaitu dengan cara menentukan data yang akan menguji kriteria. Dimana kriteria model ini akan dinilai bergantung pada *data mining goals* pada model yang akan dibangun.

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 bab ini, model sudah terbentuk dan harapannya telah memiliki kualitas baik yang terlihat dari sudut pandang analisis data. Pada tahap ini, akan dilakukan evaluasi terhadap kinerja dan kualitas model sebelum yang digunakan menentukan apakah model tersebut dapat mencapai tujuan yang ditetapkan pada fase awal (*Business Understanding*).

5.1 Evaluate Results

Pada sub bab ini bertujuan untuk mengetahui performa dari model yang telah dibangun dengan menggunakan algoritma SVM (*Support Vector Machine).*

- Evaluate menggunakan kernel Linear dengan parameter C

```
In [31]: #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)))
```

```
Model accuracy kernel Linear : 0.9985
```

```
In [32]: from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     20019
           1       1.00      1.00      1.00     20025

    accuracy                           1.00     40044
   macro avg       1.00      1.00      1.00     40044
weighted avg       1.00      1.00      1.00     40044
```

Gambar 23 Kernel Linear Parameter C

- Evaluate menggunakan kernel Polynomial dengan Parameter C

```
In [33]: #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)))
```

```
Model accuracy kernel Polynomial : 0.9751
```

```
In [34]: from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.99      0.96      0.97     20019
           1       0.96      0.99      0.98     20025

    accuracy                           0.98     40044
   macro avg       0.98      0.98      0.98     40044
weighted avg       0.98      0.98      0.98     40044
```

Gambar 24 Kernel Polynomial Parameter C

- Evaluate menggunakan Sigmoid dengan Parameter C

```
In [51]: #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_pred)))

Model accuracy kernel Sigmoid : 0.9315

In [52]: from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))

              precision    recall  f1-score   support

           0       0.93      0.93      0.93     20019
           1       0.93      0.93      0.93     20025

    accuracy                           0.93     40044
   macro avg       0.93      0.93      0.93     40044
weighted avg       0.93      0.93      0.93     40044
```


Gambar 25 Kernel Sigmoid Parameter C

Model yang dibangun menggunakan SVM sudah mendapatkan akurasi yang sesuai dengan target, dan berdasarkan evaluasi performansi model yang sudah dilakukan, bahwa model yang dibangun sudah memiliki performansi yang bagus. Dimana target sebelumnya yang diberikan oleh dosen pengampu adalah sebagai berikut:

- Precision > 0.60
- Accuracy > 0.60
- Recall > 0.65

Dan hasil classification yang didapatkan saat mengembangkan proyek ini menggunakan 2 kernel yaitu polynomial dengan hasil akurasi 0,9751, dan kernel linear kernel dengan akurasi 0,9985. Hasil *precision*, *accuracy*, dan *recall* pada kedua kernel dijelaskan sebagai berikut :

Precision

1. Menggunakan polynomial kernel :NonFraud (0,99) & Fraud(0,96)
2. Menggunakan Linear kernel: NonFraud (1,00) & Fraud(1,00)

Accuracy

1. Menggunakan polynomial kernel : NonFraud (0,96) & Fraud(0,99)
2. Menggunakan Linear kernel : NonFraud (1,00) & Fraud(1,00)

Recall

1. Menggunakan polynomial kernel : NonFraud (0,97) & Fraud(0,98)
2. Menggunakan Linear kernel : NonFraud (1,00) & Fraud(1,00)

BAB 6 DEPLOYMENT

Pada sub bab ini akan menjelaskan terkait tahapan deployment dalam melakukan prediksi Fraud Detection Train untuk dataset binary classification.

6.1. Plan Deployment

Pada tahapan plan ini, model akan dibentuk menggunakan modelling yang sesuai dengan tujuan data mining yang dibutuhkan. Model yang dihasilkan akan memerlukan dataset yang sama dengan tujuan penggunaannya. Pada kasus ini, SVM akan digunakan sesuai dengan data fraud detection tain yang diperbaharui secara real time. Data tersebut akan digunakan memprediksi keakuratan terhadap fraud yang terjadi menggunakan model yang sudah dirancang. Dataset yang digunakan harus dipastikan terlebih dahulu apakah masih memiliki missing value atau tidak, jika masih maka perlu dilakukan cleaning data terlebih dahulu. Setelah dipastikan bersih maka dataset akan diproses sesuai dengan jenis tipe datanya menggunakan model yang telah dihasilkan.

6.2. Plan Monitoring dan 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. Proyek ini dilakukan untuk menghasilkan model yang lebih kompleks di masa depan. 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

Tahapan akhir dari 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. Final report mencakup dokumen pengerjaan, file presentasi, poster dan video presentasi.

6.4. Review Project

Review project digunakan untuk menilai baik atau buruknya projek yang telah dibangun, apa yang telah selesai dan yang perlu dilakukan perbaikan. Dalam hal ini, tim terlibat dalam pengerjaan proyek dari awal hingga akhir sehingga mendapatkan pemahaman lebih detail mengenai eksplorasi data pada dataset yang digunakan, tahapan pemrosesan data untuk mendapatkan data yang siap digunakan untuk penerapan SVM.

DAFTAR PUSTAKA

[1] A. A. P. d. A. Purwarianti, "Prediksi Kinerja Penjualan Karya Musik Menggunakan Framework CRISP-DM (Studi Kasus: X Music Indonesia)," Jurnal Sarjana Institut Teknologi Bandung bidang Teknik Elektro dan Informatika, 2011.

[2] T. P. a. Y. C. I. Budiman, "Data Clustering Menggunakan Metodologi CRISP-DM Untuk Pengenalan Pola Proporsi Pelaksanaan Tridharma," J. Sist. Inf. BISNIS, 2014.