LAPORAN PRAKTIKUM
IF3270 PEMBELAJARAN MESIN

ANALISIS & EVALUASI MODEL DATA

Diajukan sebagai tugas besar Mata Kuliah IF3270 Pembelajaran Mesin

Semester II Tahun Akademik 2023/2024

Anggota Kelompok:

Jason Rivalino	13521008
M. Malik I. Baharsyah	13521029

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2023/2024

A. Hasil Analisis Data

Berikut merupakan hasil analisis data dari dataset diabetes.csv:

1. Duplicate Value

Dari hasil analisis yang ada, ditemukan bahwa terdapat jumlah data yang bernilai duplikat yaitu sebanyak 1135 data dengan total kesamaan pada baris tabel yaitu 1890 baris. Adanya kesamaan ini dapat terjadi dikarenakan terdapat beberapa pasien berbeda yang memiliki data kondisi kesehatan yang sama dengan pasien lainnya sehingga menyebabkan data menjadi duplikat.

Analisis Duplicate Value

```
# Analisis Duplicate Value
duplicate = df_train.duplicated()
print("Jumlah data yang bernilai duplikat sebanyak: ", duplicate.sum())
```

[787] ✓ 0.0s Python

```
Jumlah data yang bernilai duplikat sebanyak:  1135
```

```
# Mencari baris mana saja yang duplikat
duplicate = df_train.duplicated(keep=False)
print("Jumlah baris yang duplikat adalah: %d" % (duplicate.sum()))
print()
print("Menampilkan baris yang duplikat:")
df_train[duplicate]
```

[788] ✓ 0.0s Python

```
Jumlah baris yang duplikat adalah: 1890

Menampilkan baris yang duplikat:
```

	HighBP	HighChol	BMI	Smoker	Stroke	HeartDiseaseorAttack	PhysActivity	Fruits	Veggies	HvyAlcoholConsump	...	GenHlth	MentHlth	PhysHlth	DiffWalk	Age	Education	Income	Sex_F	Se
41464	0.0	0.0	22.0	0.0	0.0	0.0	1.0	1.0	1.0	0.0	...	2.0	0.0	0.0	0.0	8.0	5.0	7.0	1	
27504	0.0	0.0	20.0	0.0	0.0	0.0	1.0	1.0	1.0	0.0	...	1.0	0.0	0.0	0.0	9.0	6.0	8.0	1	
27383	0.0	0.0	20.0	0.0	0.0	0.0	1.0	1.0	1.0	0.0	...	1.0	0.0	0.0	0.0	2.0	6.0	8.0	0	
3635	0.0	0.0	28.0	0.0	0.0	0.0	1.0	1.0	1.0	0.0	...	1.0	0.0	0.0	0.0	6.0	6.0	8.0	1	
27364	0.0	0.0	26.0	1.0	0.0	0.0	1.0	1.0	1.0	0.0	...	2.0	0.0	0.0	0.0	5.0	5.0	8.0	1	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
10158	0.0	0.0	21.0	0.0	0.0	0.0	1.0	1.0	1.0	1.0	...	2.0	0.0	0.0	0.0	9.0	6.0	7.0	1	
22313	0.0	0.0	23.0	1.0	0.0	0.0	1.0	1.0	1.0	0.0	...	1.0	0.0	0.0	0.0	4.0	6.0	8.0	1	
19617	0.0	0.0	26.0	0.0	0.0	0.0	1.0	1.0	1.0	0.0	...	1.0	0.0	0.0	0.0	5.0	6.0	8.0	0	
15013	0.0	0.0	21.0	0.0	0.0	0.0	1.0	1.0	1.0	0.0	...	1.0	0.0	0.0	0.0	5.0	6.0	8.0	1	
6939	0.0	0.0	25.0	0.0	0.0	0.0	1.0	1.0	1.0	0.0	...	2.0	0.0	0.0	0.0	8.0	6.0	8.0	1	

1890 rows × 21 columns

2. Missing Value

Dari hasil analisis yang ada, ditemukan bahwa tidak terdapat adanya missing value sama sekali untuk keseluruhan data sehingga tidak perlu dilakukan penanganan terkait dengan missing value.

Analisis Missing Value

```
# Analisis Missing Value
missing_data = df_train.isnull().sum()
print("Jumlah missing value masing-masing kolom:")
print(missing_data)
```

✓ 0.0s

```
Jumlah missing value masing-masing kolom:
HighBP                0
HighChol              0
BMI                   0
Smoker                0
Stroke                0
HeartDiseaseorAttack  0
PhysActivity          0
Fruits                0
Veggies               0
HvyAlcoholConsump     0
AnyHealthcare         0
GenHlth               0
MentHlth              0
PhysHlth              0
DiffWalk              0
Age                   0
Education             0
Income                0
Sex_F                 0
Sex_M                 0
Diabetes              0
dtype: int64
```

3. Outlier

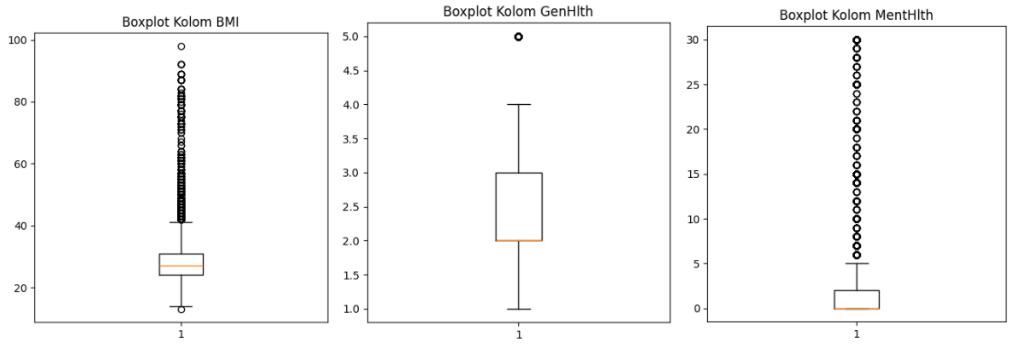
Dari hasil analisis yang ada, pengecekan outlier dilakukan untuk kolom yang bukan bernilai binary (nilainya tidak hanya 0 dan 1). Terdapat 7 kolom yang dilakukan pengecekan yaitu BMI, GenHlth, MentHlth, PhysHlth, Age, Education, dan Income dan didapatkan hasil bahwa ditemukan adanya 4 kolom dengan outlier yang cukup tinggi yaitu untuk kolom BMI (ada 1273 data), GenHlth (ada 1493 data), MentHlth (ada 4718 data), dan PhysHlth (ada 5205 data). Adanya outlier dapat terjadi karena kondisi pasien yang mengalami penyakit bisa menyebabkan data dari keadaan fisiknya berbeda dengan orang normal.

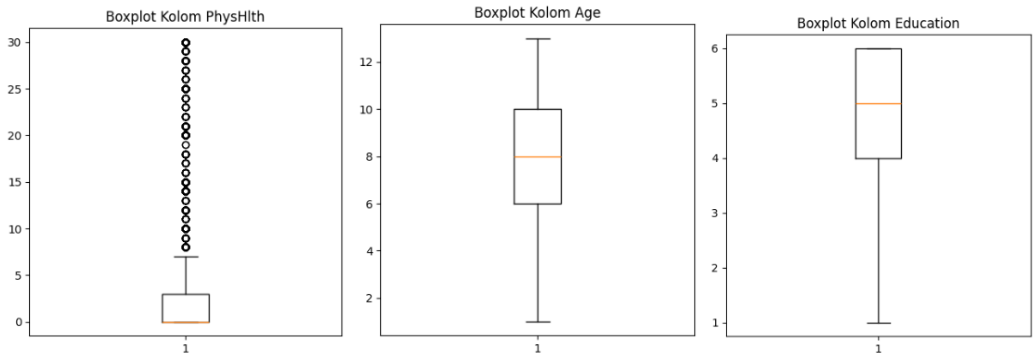
Analisis Outlier

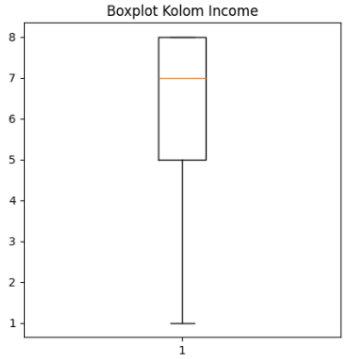
```
# Analisis Outlier Pada Kolom Numerik yang Bukan Binary
# Alasan mengapa hanya Kolom Numerik yang dipilih dan bukan binary yang diambil adalah karena binary tidak memiliki outlier
numerical_cols = df_train.select_dtypes(include=[np.number]).columns
non_binary_cols = [col for col in numerical_cols if len(df_train[col].unique()) > 2]

# Select Kolom Numerik yang Bukan Binary
df_train_outlier = df_train[non_binary_cols]

# Mencari jumlah outlier untuk setiap kolom
Q1 = df_train_outlier.quantile(0.25)
Q3 = df_train_outlier.quantile(0.75)
IQR = Q3 - Q1
outlier = ((df_train_outlier < (Q1 - 1.5 * IQR)) | (df_train_outlier > (Q3 + 1.5 * IQR))).sum()
print("Jumlah data outlier masing-masing kolom:")
print(outlier)
print()
print()

# Visualisasi data outlier dengan boxplot
print("Boxplot data outlier:")
for col in df_train_outlier.columns:
    plt.figure(figsize=(5,5))
    plt.boxplot(df_train_outlier[col])
    plt.title("Boxplot Kolom " + col)
    plt.show()
```

[790] ✓ 0.3s

```
Jumlah data outlier masing-masing kolom:
BMI          1273
GenHlth      1493
MentHlth     4718
PhysHlth     5205
Age             0
Education       0
Income          0
dtype: int64
```

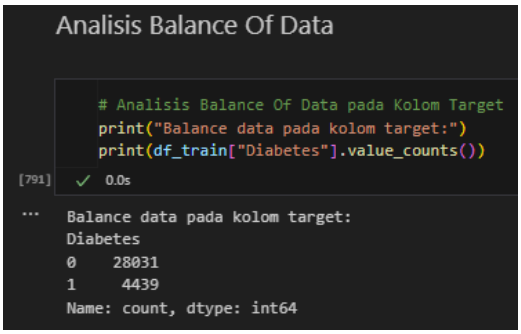






4. Balance Of Data

Dari hasil analisis yang ada, ditemukan bahwa untuk data yang ada pada atribut kolom target, data yang ada tidak seimbang dan mengalami imbalance dikarenakan data yang ada terlalu banyak yang bernilai FALSE (0) hingga 28031 data sedangkan data yang bernilai TRUE (1) hanya sebanyak 4439 data saja.



B. Penanganan dari Hasil Analisis Data

1. Duplicate Value

   Penanganan dari hasil analisis data yang ada karena banyaknya terjadi duplikat pada data, maka solusi yang dilakukan adalah dengan melakukan penambahan atribut baru yaitu ID dengan tujuan untuk membedakan setiap identifikasi berdasarkan orang secara masing-masing sehingga setiap baris akan bersifat unik dan data tidak akan mengalami duplikasi.

2. Missing Value

   Tidak perlu dilakukan penanganan karena data yang ada sudah tidak ada missing value sama sekali.

3. Outlier

   Penanganan dilakukan dengan mengisi nilai yang mengandung outlier dengan nilai kuartil atas dan kuartil bawah jika data berada diluar range interkuartil sehingga data tidak akan mengandung outlier.

4. Balance Of Data

   Penanganan dilakukan dengan menggunakan Oversampling dan Undersampling pada berbagai pemodelan untuk mendapatkan hasil akurasi data yang lebih baik.

C. Justifikasi Teknik-teknik yang Dipilih

1. Duplicate Value

   Penambahan atribut ID dilakukan dengan tujuan untuk menghilangkan duplikasi dan membuat setiap baris data memiliki sifat yang unik untuk setiap orang meskipun terdapat beberapa orang dengan kondisi kesehatan yang sama.

2. Missing Value

   Tidak terdapat missing value sehingga tidak perlu dilakukan penanganan.

3. Outlier

   Pengisian data dilakukan dengan pertimbangan bahwa nilai data yang ada harus berada dalam range kuartil sehingga tidak ada data yang memiliki nilai diluar jangkauan dan menjadi anomali dalam data sehingga untuk semua nilai yang ada diluar jangkauan menjadi diganti dengan nilai dari kuartil atas dan kuartil bawah.

4. Balance Of Data

   Penanganan dilakukan dengan mengisi nilai yang mengandung outlier dengan nilai kuartil atas dan kuartil bawah jika data berada diluar range interkuartil sehingga data tidak akan mengandung outlier.

D. Perubahan yang dilakukan pada jawaban poin 1—5 jika ada

Untuk perubahan yang diterapkan pada jawaban nomor 1-5, mungkin hanya dengan mengimplementasikan metode untuk penanganan Outlier untuk mendapatkan hasil data yang lebih baik.

E. Desain Eksperimen

Desain eksperimen dibuat dengan tujuan untuk mengidentifikasi kombinasi fitur dari data yang meliputi kondisi fisik dan kebiasaan hidup pasien, guna menemukan model yang paling efektif dalam memprediksi keberadaan penyakit diabetes. Variabel dependen dalam eksperimen ini adalah adanya penyakit diabetes, sementara variabel independennya mencakup semua fitur yang tersedia seperti tekanan darah tinggi (HighBP), kolesterol tinggi (HighChol), indeks massa tubuh (BMI), kebiasaan merokok, riwayat stroke, riwayat penyakit jantung atau serangan jantung, aktivitas fisik, kebiasaan mengkonsumsi buah dan sayuran, konsumsi alkohol berat, akses ke layanan kesehatan, kondisi kesehatan secara umum, kesehatan mental, kesehatan fisik, kesulitan berjalan, jenis kelamin, umur, pendidikan, dan pendapatan.

Strategi eksperimen yang digunakan meliputi pra-pemrosesan data (*preprocessing data*) dan membandingkan berbagai model prediktif untuk mendapat model dengan akurasi tertinggi dalam memprediksi penyakit diabetes. Skema validasi yang digunakan adalah metode k-fold cross-validation yang diaplikasikan pada data yang telah terbagi sebelumnya

menjadi data latih (*df_train*), data validasi (*df_val*), dan data tes (*df_test*). Hal tersebut dilakukan untuk memastikan bahwa model yang dikembangkan tidak mengalami *overfitting* dan memiliki akurasi yang dapat diterima.

F. Hasil Eksperimen

1. Preprocessing Data

1.1. Menghilangkan Baris Duplikat dengan Penambahan Kolom ID

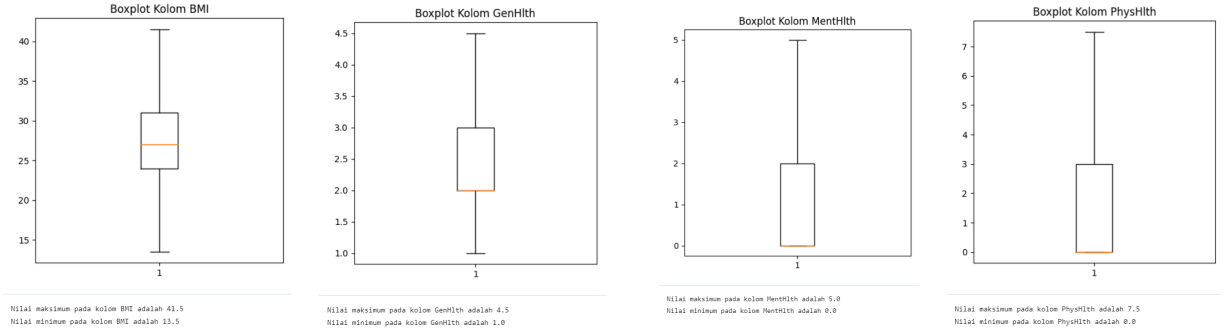
Visualize

	ID int64 0 - 50734	HighBP float64 0.0 - 1.0	HighChol float64 0.0 - 1.0	BMI float64	Smoker float64	Stroke float64	HeartDiseaseorAtt...	F
0	0	0	1	24	0	0	0	
3	3	0	0	28	1	0	0	
5	5	0	1	24	0	0	0	
6	6	0	0	20	0	0	0	
7	7	0	1	20	0	0	0	
9	9	0	0	27	1	0	0	
11	11	0	0	27	0	0	0	
12	12	1	1	30	1	0	1	
13	13	1	1	28	1	0	0	
14	14	0	1	26	1	0	0	

32470 rows, showing 10 per page Page 1 of 3247

1.2. Mengganti Nilai Outlier dengan Nilai Maksimal dari Kuartil Atas dan Kuartil Bawah

```
Batas outlier untuk setiap kolom:
Kolom BMI memiliki batas outlier: 13.5 dan 41.5
Kolom GenHlth memiliki batas outlier: 0.5 dan 4.5
Kolom MentHlth memiliki batas outlier: -3.0 dan 5.0
Kolom PhysHlth memiliki batas outlier: -4.5 dan 7.5


Boxplot data setelah perubahan nilai outlier:
```



Data setelah perubahan nilai outlier:

Visualize

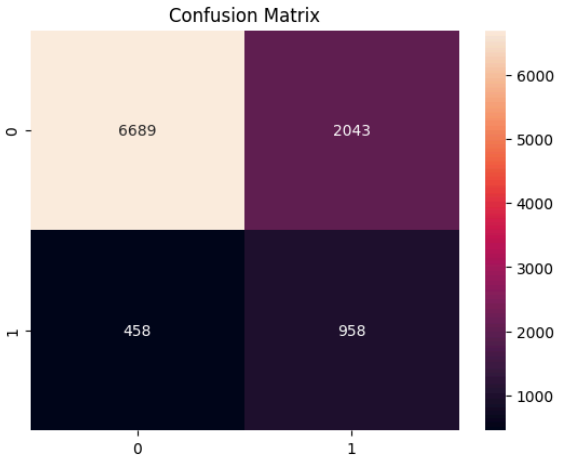
	ID int64 0 - 50734	HighBP float64 0.0 - 1.0	HighChol float64 0.0 - 1.0	BMI float64	Smoker float64	Stroke float64	HeartDiseaseorAtt...	F
167...	16749	0	1	15	1	0	0	
177..	17776	0	0	15	0	0	0	
167..	16795	0	0	15	1	0	0	
411..	41181	0	0	15	1	0	0	
32...	32002	0	0	15	1	0	0	
38...	38482	0	0	15	1	0	0	
145..	14564	0	1	15	0	0	0	
24...	24048	1	1	15	1	1	0	
6729	6729	1	1	14	1	0	0	
43...	43268	0	0	13.5	1	0	0	

32470 rows, showing 10 per page Page 3247 of 3247

1.3. Penanganan Imbalance Dataset dengan Oversampling dan Undersampling

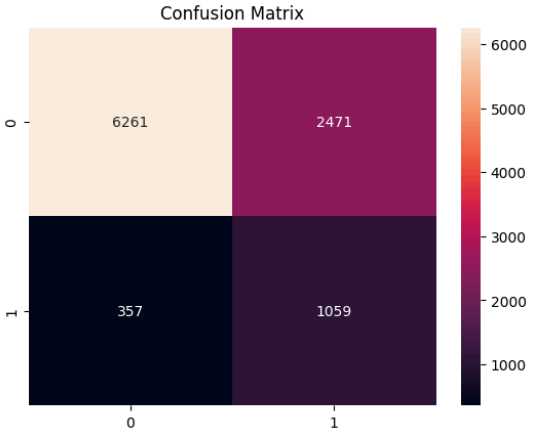
Oversampling:

```
Jumlah data setelah resampling:
(56062, 20)

Hasil akurasi dari model data adalah:  0.7535474970437525
Hasil precision dari model data adalah:  0.31922692435854716
Hasil recall dari model data adalah:  0.6765536723163842
Hasil f1 dari model data adalah:  0.4337785827484718
```



Undersampling:

```
Jumlah data setelah resampling:
(8878, 20)

Hasil akurasi dari model data adalah:  0.7213243988963343
Hasil precision dari model data adalah:  0.3
Hasil recall dari model data adalah:  0.7478813559322034
Hasil f1 dari model data adalah:  0.42822482814395474
```



2. Implementasi Model

2.1. Implementasi Model 1: Random Forest Classifier

```
Hasil akurasi dari model data test adalah:  0.8554394954670871
Hasil precision dari model data test adalah:  0.4537205081669691
Hasil recall dari model data test adalah:  0.1765536723163842
Hasil f1 dari model data test adalah:  0.2541942043721403

5-fold CV Accuracy: 0.923299159762417 ± 0.001280241473961601
5-fold CV Precision: 0.9592832002531579 ± 0.0019306587140638044
5-fold CV Recall: 0.8841216576726108 ± 0.001988606564173554
5-fold CV F1 Score: 0.9201670795150164 ± 0.0010287656536529129
```

2.2. Implementasi Model 2: XGBoostClassifier

```
Hasil akurasi dari model data test adalah:  0.861647615293654
Hasil precision dari model data test adalah:  0.5116731517509727
Hasil recall dari model data test adalah:  0.18573446327683615
Hasil f1 dari model data test adalah:  0.27253886010362693

5-fold CV Accuracy: 0.9185366298781876 ± 0.0009771787390857026
5-fold CV Precision: 0.9614343534959653 ± 0.0021387854809303366
5-fold CV Recall: 0.8720452934189871 ± 0.0020898669370082957
5-fold CV F1 Score: 0.9145567328686479 ± 0.0008764659611181304
```

2.3. Implementasi Model 3: DecisionTreeClassifier

```
Hasil akurasi dari model data test adalah:  0.82163973196689
Hasil precision dari model data test adalah:  0.38994413407821227
Hasil recall dari model data test adalah:  0.4929378531073446
Hasil f1 dari model data test adalah:  0.4354335620711166

5-fold CV Accuracy: 0.870054680810291 ± 0.003393796251262084
5-fold CV Precision: 0.8834035755856027 ± 0.0043203381652441785
5-fold CV Recall: 0.8526897212892293 ± 0.011201659872267927
5-fold CV F1 Score: 0.8677113122997377 ± 0.004442766965161241
```

G. Analisis dari Hasil Eksperimen

Berdasarkan hasil eksperimen yang telah dilakukan, berikut ini adalah analisis untuk setiap pendekatan dan model yang digunakan:

Oversampling: Melalui teknik SMOTE, data kelas minoritas (kasus diabetes) diperbanyak untuk menciptakan keseimbangan dengan kelas mayoritas. Model Logistic Regression yang dilatih pada data ini menghasilkan skor F1 yang lebih tinggi (0.4338) dibandingkan dengan pendekatan lain, menunjukkan peningkatan dalam mengidentifikasi kelas minoritas.

Undersampling: Teknik ini mengurangi jumlah data pada kelas mayoritas untuk menyamakan dengan kelas minoritas, namun menghasilkan skor F1 yang lebih rendah (0.4282) dibandingkan dengan oversampling, menunjukkan bahwa mengurangi jumlah data dapat mengurangi kapasitas model dalam menggeneralisasi data yang belum dilihat.

1. **Random Forest Classifier pada Data Oversampling:**

   Model ini memberikan hasil yang lebih baik dalam hal accuracy (0.8554) dibandingkan dengan Logistic Regression, namun skor F1 dan recall tetap rendah. Hal ini menunjukkan bahwa model mungkin baik dalam mengklasifikasikan kelas mayoritas tetapi kurang efektif untuk kelas minoritas (diabetes positif).

2. **XGBoost Classifier pada Data Oversampling:**

   Dengan skor F1 yang lebih tinggi (0.2725) dibandingkan dengan Random Forest, XGBoost menunjukkan peningkatan dalam hal menyeimbangkan precision dan recall. Namun, nilai ini masih menunjukkan bahwa ada kesulitan signifikan dalam mengidentifikasi kasus diabetes secara akurat.

3. **DecisionTreeClassifier pada Data Oversampling:**

   Model ini memberikan skor yang paling rendah jika dibandingkan dengan skor lainnya sehingga model ini kurang efektif untuk digunakan.

Hal lain yang didapat juga bahwa dengan penerapan 5 K-Folds, dapat membantu untuk meningkatkan nilai dari akurasi, presisi, recall, dan juga F1-Score pada data yang bertujuan untuk membantu mendapatkan data dengan hasil yang lebih baik.

H. Kesimpulan

Kesimpulan yang didapat dari praktikum ini adalah sebagai berikut:

1. Beberapa data yang didapat dari awal masih mengalami beberapa kekurangan yang perlu dilakukan tuning untuk meningkatkan nilai akurasi data agar dapat menjadi lebih baik. Beberapa masalah yang mungkin terjadi yaitu duplicate value, missing value, outlier, dan imbalance
2. Penangan dapat dilakukan dengan berbagai cara mulai dari mengisi dan mengganti data, melakukan overscaling dan undersampling, menambah atribut baru, ataupun cara lainnya untuk mendapatkan bentuk prediksi data terbaik
3. Penggunaan model dengan XGBoost dapat memberikan hasil yang paling baik dalam menentukan prediksi data untuk kolom jika dibandingkan dengan RandomForest dan DecisionTree

I. Pembagian Tugas/kerja per Anggota Kelompok

Nama	NIM	Pembagian Kerja
Jason Rivalino	13521008	Nomor 1-3, Nomor 6-7, Laporan
M. Malik I. Baharsyah	13521029	Nomor 4-5, Nomor 6-7, Laporan