LAPORAN PRAKTIKUM
MACHINE LEARNING

Disusun Oleh:
Michael Utama (13521137)
Kandida Edgina Gunawan (13521155)

DAFTAR ISI

I. Hasil Analisis Data

a. Penjelasan Dataset

Dataset yang diberikan (*diabetes.csv*) merupakan dataset yang telah dimodifikasi dari Diabetes Health Indicators Dataset yang berisi kumpulan indikator individu yang diperoleh dari survei untuk kasus diabetes. Dataset ini digunakan untuk melakukan prediksi apakah seorang individu mengalami diabetes atau tidak berdasarkan faktor-faktor tertentu yang diketahui.

Dataset terdiri dari 20 kolom dan 50.736 baris dengan deskripsi singkat setiap kolomnya sebagai berikut:

1. HighBP: Memiliki tekanan darah tinggi (BP: Blood Pressure) atau tidak
2. HighChol: Kolesterol tinggi atau tidak
3. BMI: Besaran Body Mass Index
4. Smoker: Perokok atau bukan perokok
5. Stroke: Pernah mengalami struk atau tidak
6. HeartDiseaseorAttack: Memiliki riwayat penyakit antara jantung koroner dan serangan jantung atau tidak sama sekali
7. PhysActivity: Aktif secara fisik dalam 30 hari terakhir atau tidak
8. Fruits: Mengonsumsi buah setiap hari atau tidak
9. Veggies: Mengonsumsi sayur setiap hari atau tidak
10. HvyAlcoholConsump: Peminum berat alkohol atau bukan
11. AnyHealthcare: Memiliki perlindungan kesehatan atau tidak, contohnya memiliki asuransi kesehatan
12. GenHtlth: Evaluasi mandiri terhadap kesehatan, skala 1-5 (1: Sangat baik, 2: Cukup Baik, 3: Baik, 4: Biasa saja, 5: Buruk)
13. MentHlth: Jumlah hari keadaan mental buruk dalam 30 hari terakhir (skala 0-30 hari)
14. PhysHlth: Jumlah hari keadaan fisik buruk dalam 30 hari terakhir (skala 0-30 hari)
15. DiffWalk: Memiliki kesulitan berjalan atau menaiki tangga
16. Sex: (M) Male atau (F) Female
17. Age: 13 kategori umur (1: 18-24 tahun, 9: 60-64 tahun, 13: 80 tahun ke atas)
18. Education: Level edukasi skala 1-6 (1: Tidak pernah sekolah atau hanya TK, 2: SD, dst)
19. Income: Skala pendapatan 1-8
20. Diabetes: Apakah mengalami diabetes atau tidak (Kolom target)

b. Data types

```
HighBP                  float64
HighChol                float64
BMI                     float64
Smoker                  float64
Stroke                  float64
HeartDiseaseorAttack    float64
PhysActivity            float64
Fruits                  float64
Veggies                 float64
HvyAlcoholConsump       float64
AnyHealthcare           float64
GenHlth                 float64
MentHlth                float64
PhysHlth                float64
DiffWalk                float64
Sex                      object
Age                     float64
Education               float64
Income                  float64
Diabetes                   bool
dtype: object
```

Berdasarkan dataset yang diberikan, dari 20 kolom dataset tersebut, 18 di antaranya bertipe float24, sedangkan 2 di antaranya, yaitu Sex dan Diabetes, masing-masing bertipe object dan bool. Pada kolom Sex, terdapat 2 kemungkinan nilai, yaitu 'F' (menyatakan *Female*) dan 'M' (menyatakan *Male*). Pada kolom Diabetes yang bertipe bool juga hanya dijumpai 2 kemungkinan nilai, yaitu *True* (menyatakan individu mengalami diabetes) dan *False* (menyatakan individu tidak mengalami diabetes.

c. Duplicate Values

```
## Duplicated Rows
print(f"Duplicated rows: {df_train.duplicated().sum()}")

## Duplicated Independent Instances
print(f"Duplicated independent instances: {df_train.drop('Diabetes', axis = 1).duplicated().sum()}")
```

```
Duplicated rows: 1135
Duplicated independent instances: 1193
```

Berdasarkan eksekusi program di atas, ditemukan terdapat 1.135 *duplicate rows* pada training set. Selain itu, ditemukan 1.193 baris data yang memiliki *duplicate independent features* (seluruh *independent features*-nya memiliki nilai yang tepat sama) dan setelah diselidiki lebih lanjut, sebagian diantaranya bersifat *contradictory* karena memiliki 2 baris atau lebih yang mempunyai *independent features* dengan nilai yang tepat sama, namun menghasilkan *target variable* yang berbeda.

d. Missing Value

```
1  data.isna().sum()
```

```
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
Sex                     0
Age                     0
Education               0
Income                  0
Diabetes                0
dtype: int64
```

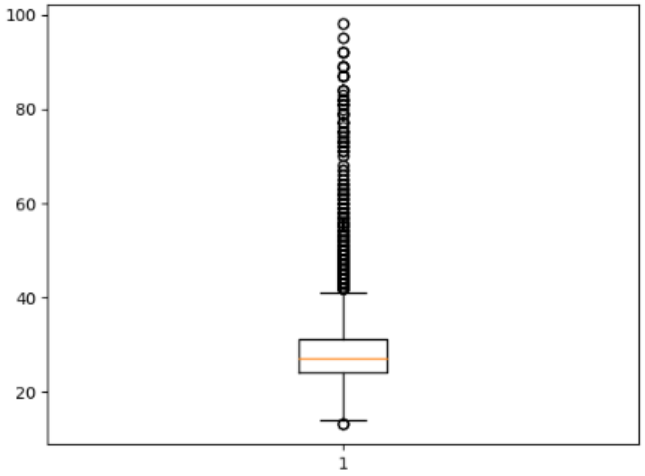
Berdasarkan eksekusi kode di atas, setiap kolom data tidak memiliki *missing value*.

e. Outlier

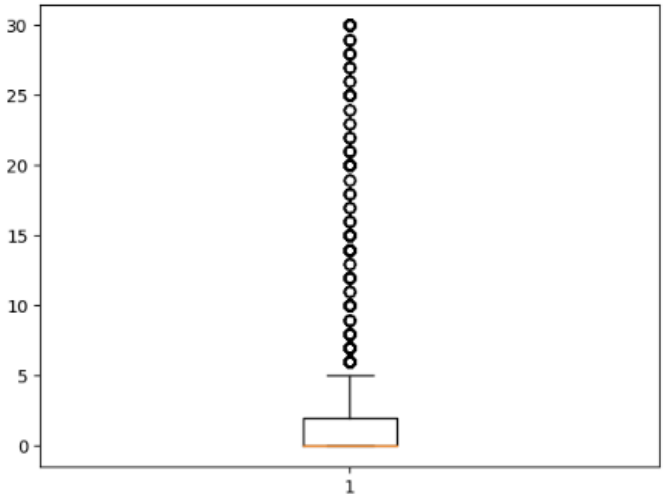
Prinsip *outlier* hanya dimiliki oleh *non-categorical variable*. Oleh karena itu, perlu ditinjau sebelumnya fitur-fitur apa saja pada dataset yang termasuk *categorical variable* dan *non-categorical variable.* Berdasarkan *data type* dari setiap fitur di atas, fitur *Diabetes* dan *Sex* yang tidak bertipe *numerical* pastilah dikategorikan dalam *categorical variable*. Untuk fitur-fitur lainnya, seperti HighBP, HighChol, BMI, Smoker, Stroke, HeartDiseaseorAttack, Fruits, Veggies, HvyAlcoholConsump, dan AnyHealthcare, walaupun bertipe float24, fitur-fitur tersebut sebenarnya merupakan *categorical feature* karena menyatakan apakah suatu kondisi benar atau tidak dengan nilainya telah di-*encode* dengan *label encoding* 0 atau 1. Fitur seperti GenHlth dan Education juga merupakan *categorical variable* dalam bentuk ordinal yang telah di-*encode* nilainya dengan menggunakan *label encoding*. Untuk fitur *income* dan *age,* telah dikategorisasi *binning* untuk mengubah nilai dari fiturnya ke dalam kategori dengan *range* nilai tertentu sesuai *bin*-nya. Nilai dari *income* dan *age* juga telah di-*encode* dengan *label encoding*.

Fitur seperti BMI, MentHlth, dan PhysHlth dikategorikan ke dalam *non-categorical variable* sehingga fitur-fitur tersebut dapat dicari *outlier*-nya dengan menggunakan prinsip interkuartil. Berikut ini visualisasi *outlier* dari fitur-fitur tersebut dengan menggunakan *boxplot*:

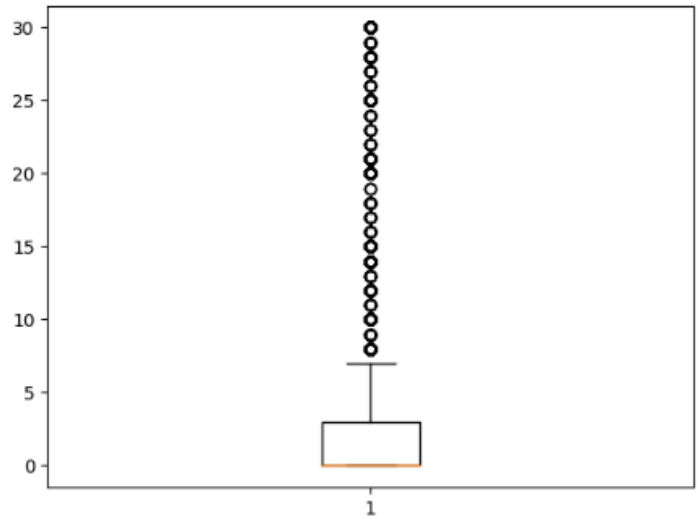
1. BMI



2. MentHlth



3. PhysHlth



f. Balance of Data

```
1  data['Diabetes'].value_counts()
```

```
Diabetes
False    43790
True      6946
Name: count, dtype: int64
```

Berdasarkan eksekusi kode di atas, dapat dilihat bahwa dataset yang diberikan *imbalanced* karena terdapat 2 kemungkinan hasil target, tetapi pada *dataset* kemunculan target *True* dan *False*-nya tidak seimbang (43.790 kemunculan *False* dan 6.946 kemunculan *True*). Hal ini jika dibiarkan dapat menyebabkan model yang diciptakan menjadi bias, lebih cenderung kepada *majority class*, yaitu *False*.

II. Penanganan dari Hasil Analisis Data dan Justifikasi Teknik yang Dipilih

a. Duplicate Values

Untuk kasus duplicate rows, setiap duplicated rows akan di-*drop* dan menyisakan *first occurence* dari setiap *row* yang terduplikasi tersebut. Hal ini dilakukan untuk mencegah munculnya bias akibat duplikasi, namun dengan masih mempertahankan *instance* unik yang ada untuk menambah *knowledge* dalam proses learning. Untuk kasus *contradictory rows*, dilakukan proses *dropping* untuk setiap *row* yang terlibat. Hal ini dilakukan karena kita tidak dapat menentukan *row* mana yang masih relevan dan yang mana yang tidak. Selain itu jumlah *row* yang mengalami *contradictory* juga relatif kecil jika dibandingkan dengan ukuran dataset yang dimiliki.

b. Outlier

Kasus *outlier* ditangani dengan menggunakan teknik *winsorization*. Teknik *winsorization* merupakan teknik untuk mengganti *extreme values* dari statistikal data untuk mengurangi efek dari *outlier* pada proses *learning* data. Pada kolom BMI, digunakan teknik 90% *winsorization*, yaitu dengan mengganti data dari top 5% dan bottom 5% dengan 95th percentile dan 5th percentile berturut-turut. Alasan konsiderasi pengambilan percentile ini adalah karena teknik *winsorization* yang populer adalah 90% *winsorization* dan 99% *winsorization*. Akan tetapi, setelah dilakukan uji coba, teknik 99% *winsorization* kurang dapat mengurangi efek dari *outlier* sehingga terpilihlah teknik 90% *winsorization.*

Pada kolom MentHlth dan PhysHlth tidak dilakukan perubahan terhadap instance yang terindikasi *outlier*. Hal ini dilakukan karena penulis menganggap mengubah atau menghapus *outlier* pada kedua kolom tersebut akan menyebabkan *knowledge* penting hilang dalam proses *learning*.

c. Encoding

Berdasarkan dataset, terdapat 2 kolom yang masih bernilai non numerik, yaitu Sex (bertipe object) dan Diabetes (bertipe boolean). Perlu dilakukan encoding untuk feature-feature tersebut karena beberapa model machine learning mengharuskan input dan output dalam bentuk numerik. Kolom Sex memiliki 2 kemungkinan nilai yaitu F atau M. Nilai F dan M tidak memiliki ordinalitas tertentu yang menyatakan M > F ataupun F < M. Label Encoding biasanya digunakan pada feature ketika terdapat ordinalitas tertentu pada feature tersebut, sedangkan One Hot Encoding digunakan pada feature nominal. Akan tetapi, karena pada kolom sex hanya terdapat 2 kemungkinan nilai, yaitu F dan M yang mana jika diubah ke dalam Label Encoding akan menghasilkan 0 atau 1 untuk masing-masing F dan M, begitupun sebenarnya untuk One Hot Encoding . Akan tetapi pada one hot encoding diperlukan 2 kolom untuk masing-masing menyatan F dan M. Dengan pertimbangan dimensionalitas dan efektifitas, label encoding sudah cukup untuk merepresentasikan kolom Sex dengan baik

Kolom Diabetes memiliki 2 kemungkinan nilai yaitu True atau False dan diberlakukan Label Encoding dengan alasan yang serupa dengan kolom Sex di atas.

d. Balance of Data

Imbalanced dataset ditangani dengan mengubah evaluation metric, menggunakan data level methods dan algorithm level methods.

Evaluation metric yang digunakan tidak hanya accuracy karena akan memberikan nilai besar jika prediksi selalu salah, namun juga recall, precision, dan F1-Score.

Penanganan dengan data level methods dilakukan dengan oversampling, agar tidak membuang data untuk kelas False.

Penanganan dengan algorithm level methods dilakukan dengan memberikan cost untuk misclassification tiap kelas, memindah threshold untuk mendapat precision-recall tradeoff yang baik, dan ensemble classifier.

e. Remove Feature

Pada dataset, terdapat 20 feature yang belum tentu semuanya memberikan kontribusi yang besar pada hasil prediksi Diabetes. Beberapa fitur bisa saja membuat hasil prediksi menjadi bias. Oleh karena itu, penulis menghapus 5 fitur yang memiliki nilai kepentingan paling rendah dalam proses prediksi target Diabetes. Untuk menentukan fitur-fitur dengan *score* tertinggi (paling penting dalam prediksi Diabetes), digunakan SelectKBest, yang merupakan salah satu metode *feature selection*. Jenis SelectKBest yang digunakan adalah SelectKBest dengan menggunakan *statistical test* chi-squared test.

Berikut ini merupakan hasil perhitungan *importance score* dari setiap fitur, terurut dari fitur dengan *importance score* terbesar ke yang terkecil:

	feature object PhysHlth 5.3% BMI 5.3% 17 others 89.5%	score float64 3.14499550051101...
13	PhysHlth	36538.77657
2	BMI	5833.470821
12	MentHlth	5514.082077
16	Age	4157.334919
11	GenHlth	3674.869728
0	HighBP	3526.32409
14	DiffWalk	2565.680802
1	HighChol	2142.395871
18	Income	2044.613593
5	HeartDiseaseorAt...	1670.499514

	feature object PhysHlth 5.3% BMI 5.3% 17 others 89.5%	score float64 3.14499550051101...
4	Stroke	619.5308863
9	HvyAlcoholConsu...	545.5124937
17	Education	316.9089854
3	Smoker	314.0793123
6	PhysActivity	311.0863216
15	Sex	247.4220472
8	Veggies	39.92536937
7	Fruits	22.8696965
10	AnyHealthcare	3.144995501

Dengan demikian 5 fitur dengan *importance score* terendah, yaitu PhysActivity, Sex, Veggies, Fruits, dan AnyHealthcare akan didrop dari dataset.

f. Scaling

Proses *scaling* perlu dilakukan pada dataset untuk memastikan tidak terdapat fitur tertentu yang mendominasi proses *learning* karena nilai yang dimilikinya besar. Dilakukan proses *scaling* pada dataset dengan metode Standard Scaler yang mengubah nilai atribut dalam *range* 0 dan 1. Metode Standard Scaler digunakan karena berbeda dengan *normalization* yang sangat dipengaruhi oleh *outlier*, Standard Scaler merupakan *standardization* yang hampir tidak dipengaruhi oleh *outlier*. Hal ini dinilai penting oleh penulis, mengingat terdapat kolom-kolom yang sebenarnya memiliki nilai *outlier* tapi tidak mengalami perubahan tertentu.

III. Desain Eksperimen

a. Tujuan Eksperimen

Eksperimen bertujuan untuk melakukan prediksi apakah seseorang pernah mengalami diabetes atau tidak, berdasar data kesehatan yang diberikan.

b. Variabel dependen dan independen

Variabel dependen: Diabetes

Variabel independen:

1. HighBP Memiliki tekanan darah tinggi (BP: Blood Pressure) atau tidak
2. HighChol Kolesterol tinggi atau tidak
3. BMI: Besaran Body Mass Index
4. Smoker: Perokok atau bukan perokok
5. Stroke: Pernah mengalami struk atau tidak
6. HeartDiseaseorAttack: Memiliki riwayat penyakit antara jantung koroner dan serangan jantung atau tidak sama sekali
7. PhysActivity: Aktif secara fisik dalam 30 hari terakhir atau tidak
8. Fruits: Mengonsumsi buah setiap hari atau tidak
9. Veggies: Mengonsumsi sayur setiap hari atau tidak
10. HvyAlcoholConsump: Peminum berat alkohol atau bukan
11. AnyHealthcare: Memiliki perlindungan kesehatan atau tidak, contohnya memiliki asuransi kesehatan
12. GenHtlth: Evaluasi mandiri terhadap kesehatan, skala 1-5 (1: Sangat baik, 2: Cukup Baik, 3: Baik, 4: Biasa saja, 5: Buruk)
13. MentHlth: Jumlah hari keadaan mental buruk dalam 30 hari terakhir (skala 0-30 hari)
14. PhysHlth: Jumlah hari keadaan fisik buruk dalam 30 hari terakhir (skala 0-30 hari)
15. DiffWalk: Memiliki kesulitan berjalan atau menaiki tangga
16. Sex: (M) Male atau (F) Female
17. Age: 13 kategori umur (1: 18-24 tahun, 9: 60-64 tahun, 13: 80 tahun ke atas)
18. Education: Level edukasi skala 1-6 (1: Tidak pernah sekolah atau hanya TK, 2: SD, dst)
19. Income: Skala pendapatan 1-8

c. Strategi Eksperimen
Strategi yang digunakan untuk eksperimen ini adalah *One Factor at A Time*. Baseline yang digunakan mengikuti baseline model. Faktor yang akan diubah termasuk, tetapi tidak terbatas pada

1. Penalty model logistic regression
2. Weight tiap fitur (balanced, default, custom)
3. Model atau gabungan model yang digunakan

d. Skema Validasi
Skema validasi yang digunakan adalah *Hold-out Validation*. Dataset dibagi menjadi data latih, data validasi, dan data tes. Eksperimen dilakukan menggunakan data validasi, dan model terbaik yang dihasilkan diuji menggunakan data tes.

IV. Perubahan yang dilakukan

Strategi yang digunakan pada eksperimen adalah *grid search*, bukan *one factor at a time*. Hal tersebut dilakukan karena terdapat library bagian sklearn yang dapat membantu melakukan *grid search*, sehingga mempercepat iterasi eksperimen.

V. Hasil Eksperimen

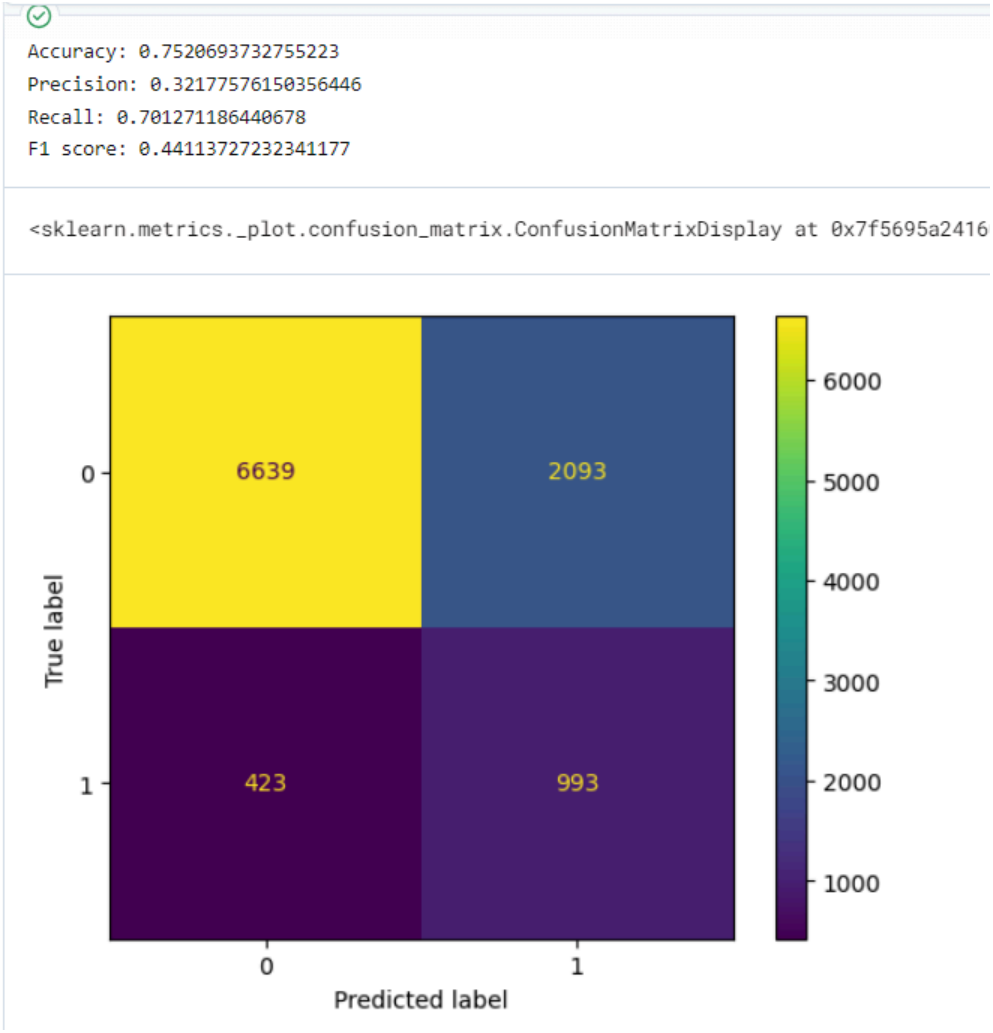
```
params = {
    'C': [1, 10, 0.1],
    'class_weight': [None, 'balanced'],
    'solver': ['lbfgs', 'newton-cg', 'saga'],
    'penalty': ['l1', 'l2', 'elasticnet', None]
}

model_logreg = LogisticRegression()

X_train_val = pd.concat([X_train, X_val])
y_train_val = pd.concat([y_train, y_val])

split_index = [-1 if x == y else 0 for x, y in zip(X_train_val.index, range(X_train_val.shape[0]))]

pds = PredefinedSplit(test_fold = split_index)

grid = GridSearchCV(model_logreg, params, scoring=metrics.make_scorer(metrics.f1_score, average='binary', pos_label=1)

grid.fit(X_train_val, y_train_val)
```

Gambar 5.1 Parameter yang digunakan dalam eksperimen

```
grid.best_estimator_
```

```
LogisticRegression
LogisticRegression(C=0.1, penalty='l1', solver='saga')
```

Gambar 5.2 Model terbaik dari hasil *Grid Search*



Gambar 5.3 Pengujian model terbaik menggunakan df_test

VI. Analisis dari Hasil Eksperimen

Beberapa model digunakan dalam eksperimen: logistic regression, k nearest neighbor, random forest, dan ensemble menggunakan stacking classifier. Pengujian baseline tiap model menunjukkan bahwa logistic regression memiliki nilai f1 paling baik.

Hyperparameter yang dituning dalam eksperimen ini adalah regularization strength, class_weight, solver, dan penalty yang digunakan, dengan f1 score sebagai . Hasil pengujian menunjukkan variasi antara pemilihan hyperparameter kecil. Selain itu, terdapat tuning untuk threshold yang dilakukan setelah menemukan model terbaik, untuk mendapatkan tradeoff precision-recall yang baik.

VII. Kesimpulan

Eksperimen dalam membentuk sebuah model *machine learning* terdiri atas tahap perencanaan dan pelaksanaan eksperimen. Tahap perencanaan meliputi pemahaman dataset, melakukan analisis data, mencari permasalahan pada data dan merencanakan penanganannya (data duplikat, outlier, imbalanced data, encoding, scaling inkonsisten,

contradictory data, dll), dan membuat desain eksperimen. Tahap pelaksanaan meliputi preprocessing data dan training model.

Preprocessing data memiliki pengaruh besar, terutama karena dataset memiliki banyak masalah. *Preprocessing* yang kami lakukan adalah menghilangkan nilai duplikat dan kontradiksi, normalisasi outlier dengan winsor, *oversampling* untuk *imbalanced* data, pemilihan fitur sesuai nilai korelasi, dan *scaling* data.

Eksperimen dilakukan dengan metode *grid search*. Hyperparameter yang diubah selama proses eksperimen ini tidak menghasilkan peningkatan nilai f1 yang signifikan.

VIII. Pembagian Tugas

NIM	Nama	Pembagian Tugas
13521137	Michael Utama	Baseline model, desain eksperimen, training model
13521155	Kandida Edgina Gunawan	Dataset, analisis data, encoding, preprocessing