CANCEL

REVENUE DECRESES

DANGER!

HOTEL RESERVATION ANALYSIS

OUR TEAM

SAEFUL FIKRI

5003211049

NAZIA MAHMUDAH

5003211157

GALIH FITRIATMO

5003211087

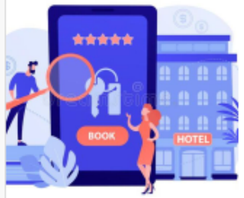
TABLE OF CONTENT

PERMASALAHAN

Dalam 1 tahun terakhir, reservasi Hotel telah mengalami peningkatan yang signifikan dalam tingkat **pembatalan** yang menyebabkan Hotel menghadapi berbagai tantangan. Seperti berkurangnya **pendapatan** dan **kamar hotel** yang **kurang termanfaatkan**. **Alasan** umum **pembatalan** termasuk perubahan rencana, konflik penjadwalan, dll. Hal ini sering kali dipermudah dengan adanya pilihan untuk melakukannya **secara gratis** atau lebih disukai dengan **biaya rendah** yang bermanfaat bagi tamu hotel. Namun hal ini merupakan faktor yang kurang diinginkan dan mungkin **mengurangi pendapatan hotel**.

Oleh karena itu, prioritas utama bagi Hotel adalah **mengurangi tingkat pembatalan** yang akan **meningkatkan efisiensi** dalam menghasilkan **pendapatan**. Penelitian ini berfokus pada **analisis pembatalan pemesanan hotel** dan **faktor-faktor** yang mempengaruhi pembatalan.

DATA



Hotel Reservations Dataset

Can you predict if customer is going to cancel the reservation ?

kaggle.com

“Hotel Reservations Dataset”

- Sumber : https://www.kaggle.com/datasets/ahsan81/hotel-reservations-classification-dataset
- Variabel: Data terdiri dari 19 variabel dan 36275 sampel.

```
df.dtypes
```

```
Booking_ID                            object
no_of_adults                           int64
no_of_children                         int64
no_of_weekend_nights                   int64
no_of_week_nights                      int64
type_of_meal_plan                     object
required_car_parking_space             int64
room_type_reserved                    object
lead_time                              int64
arrival_year                           int64
arrival_month                          int64
arrival_date                           int64
market_segment_type                   object
repeated_guest                         int64
no_of_previous_cancellations           int64
no_of_previous_bookings_not_canceled   int64
avg_price_per_room                   float64
no_of_special_requests                 int64
booking_status                        object
dtype: object
```

PRE-PROCESSING DATA

Missing Value | Duplicated Data | Data Type | **Outlier**

```
Booking_ID                              0
no_of_adults                            0
no_of_children                          0
no_of_weekend_nights                    0
no_of_week_nights                       0
type_of_meal_plan                       0
required_car_parking_space              0
room_type_reserved                      0
lead_time                               0
arrival_year                            0
arrival_month                           0
arrival_date                            0
market_segment_type                     0
repeated_guest                          0
no_of_previous_cancellations            0
no_of_previous_bookings_not_canceled    0
avg_price_per_room                      0
no_of_special_requests                  0
booking_status                          0
```

MISSING VALUE

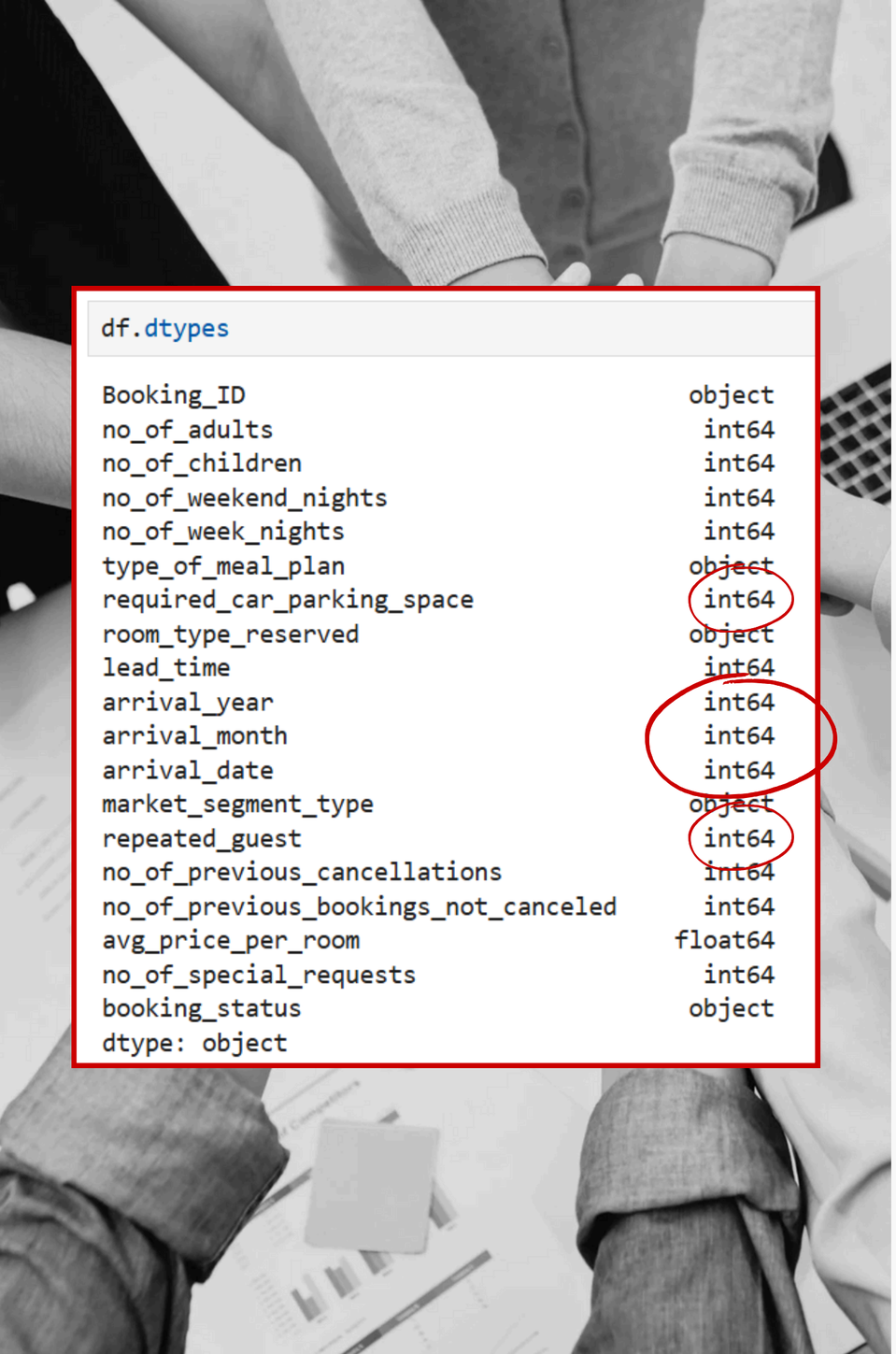
Tidak ada Missing Value

```
sum(df.duplicated())
```

```
0
```

DUPLICATED DATA

Tidak ada Duplicated Data



```
df.dtypes
```

```
Booking_ID                            object
no_of_adults                           int64
no_of_children                         int64
no_of_weekend_nights                   int64
no_of_week_nights                      int64
type_of_meal_plan                     object
required_car_parking_space             int64
room_type_reserved                    object
lead_time                              int64
arrival_year                           int64
arrival_month                          int64
arrival_date                           int64
market_segment_type                   object
repeated_guest                         int64
no_of_previous_cancellations           int64
no_of_previous_bookings_not_canceled   int64
avg_price_per_room                   float64
no_of_special_requests                 int64
booking_status                        object
dtype: object
```

TIPE DATA

```
#Merubah tipe data
df['required_car_parking_space'] = data['required_car_parking_space'].astype('str')
df['arrival_year'] = data['arrival_year'].astype('str')
df['arrival_month'] = data['arrival_month'].astype('str')
df['arrival_date'] = data['arrival_date'].astype('str')
df['repeated_guest'] = data['repeated_guest'].astype('str')
df.dtypes
```

```
Booking_ID                            object
no_of_adults                           int64
no_of_children                         int64
no_of_weekend_nights                   int64
no_of_week_nights                      int64
type_of_meal_plan                     object
required_car_parking_space            object
room_type_reserved                    object
lead_time                              int64
arrival_year                          object
arrival_month                         object
arrival_date                          object
market_segment_type                   object
repeated_guest                        object
no_of_previous_cancellations           int64
no_of_previous_bookings_not_canceled   int64
avg_price_per_room                   float64
no_of_special_requests                 int64
booking_status                        object
```

SUMMARY STATISTICS

Statistika Deskriptif | Visualisasi

STATISTIKA DESKRIPTIF

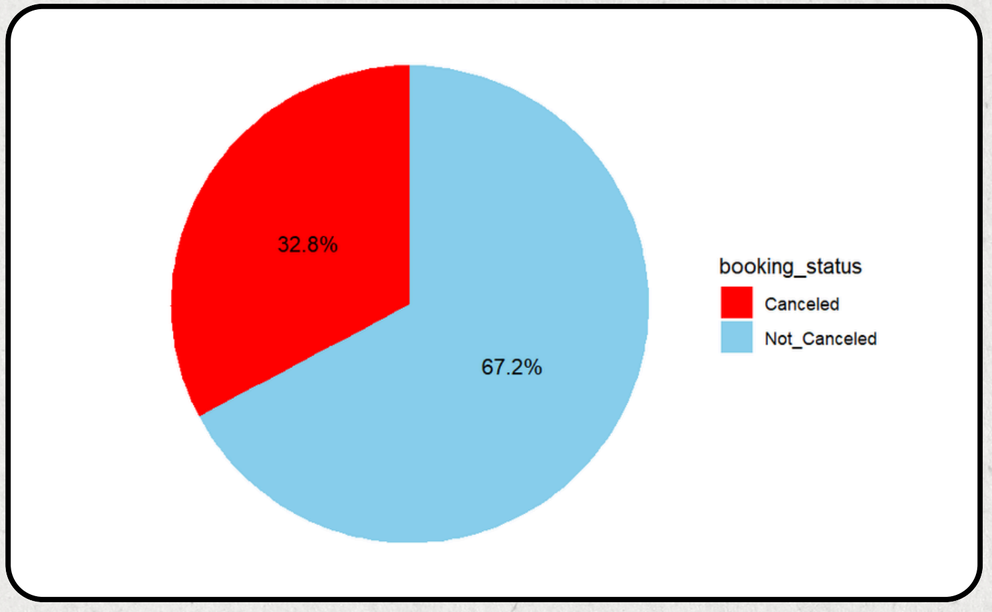
NUMERIK

	no_of_adults	no_of_children	no_of_weekend_nights	no_of_week_nights	lead_time	no_of_previous_cancellations	no_of_previous_bookings_not_canceled	avg_price_per_room	no_of_special_requests
count	36275.000000	36275.000000	36275.000000	36275.000000	36275.000000	36275.000000	36275.000000	36275.000000	36275.000000
mean	1.844962	0.105279	0.810724	2.204300	85.232557	0.023349	0.153411	103.423539	0.619655
std	0.518715	0.402648	0.870644	1.410905	85.930817	0.368331	1.754171	35.089424	0.786236
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	2.000000	0.000000	0.000000	1.000000	17.000000	0.000000	0.000000	80.300000	0.000000
50%	2.000000	0.000000	1.000000	2.000000	57.000000	0.000000	0.000000	99.450000	0.000000
75%	2.000000	0.000000	2.000000	3.000000	126.000000	0.000000	0.000000	120.000000	1.000000
max	4.000000	10.000000	7.000000	17.000000	443.000000	13.000000	58.000000	540.000000	5.000000

KATEGORIK

	Booking_ID	type_of_meal_plan	required_car_parking_space	room_type_reserved	arrival_year	arrival_month	arrival_date	market_segment_type	repeated_guest	booking_status
count	36275	36275	36275	36275	36275	36275	36275	36275	36275	36275
unique	36275	4	2	7	2	12	31	5	2	2
top	INN00001	Meal Plan 1	0	Room_Type 1	2018	10	13	Online	0	Not_Canceled
freq	1	27835	35151	28130	29761	5317	1358	23214	35345	24390

STATISTIKA DESKRIPTIF




```
Confusion Matrix for DecisionTree :
                Reference
Prediction      Canceled Not_Canceled
  Canceled             0            0
  Not_Canceled      3593         7290
F1 Score for DecisionTree : NA
```

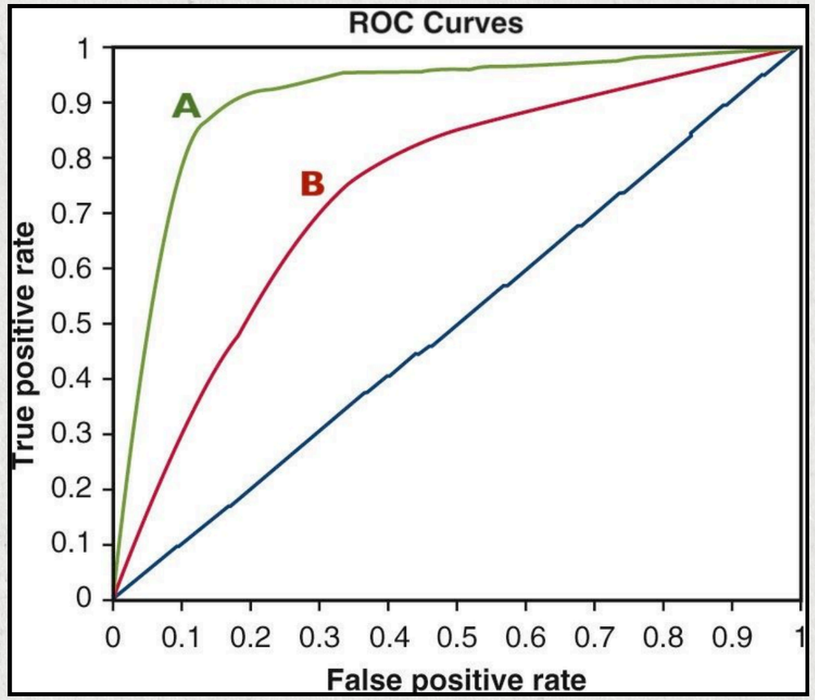
Model condong hanya mampu mengklasifikasikan kelas yang majoritas

IMBALANCE DATA !

SOLVING IMBALANCE DATA

ROC CURVE!

Metrik AUC-ROC robust untuk data yang tidak seimbang karena metrik ini mengevaluasi kinerja model di semua ambang batas klasifikasi dan tidak secara langsung dipengaruhi oleh ketidakseimbangan proporsi kelas. Classifier yang dihasilkan dapat divisualisasikan pada kurva ROC dan recall terhadap 1-precision untuk memilih classifier terbaik (Hajo Holzmann & Bernhard Klar, 2024)



LET’S DIVE DEEPER!

FEATURE SELECTION

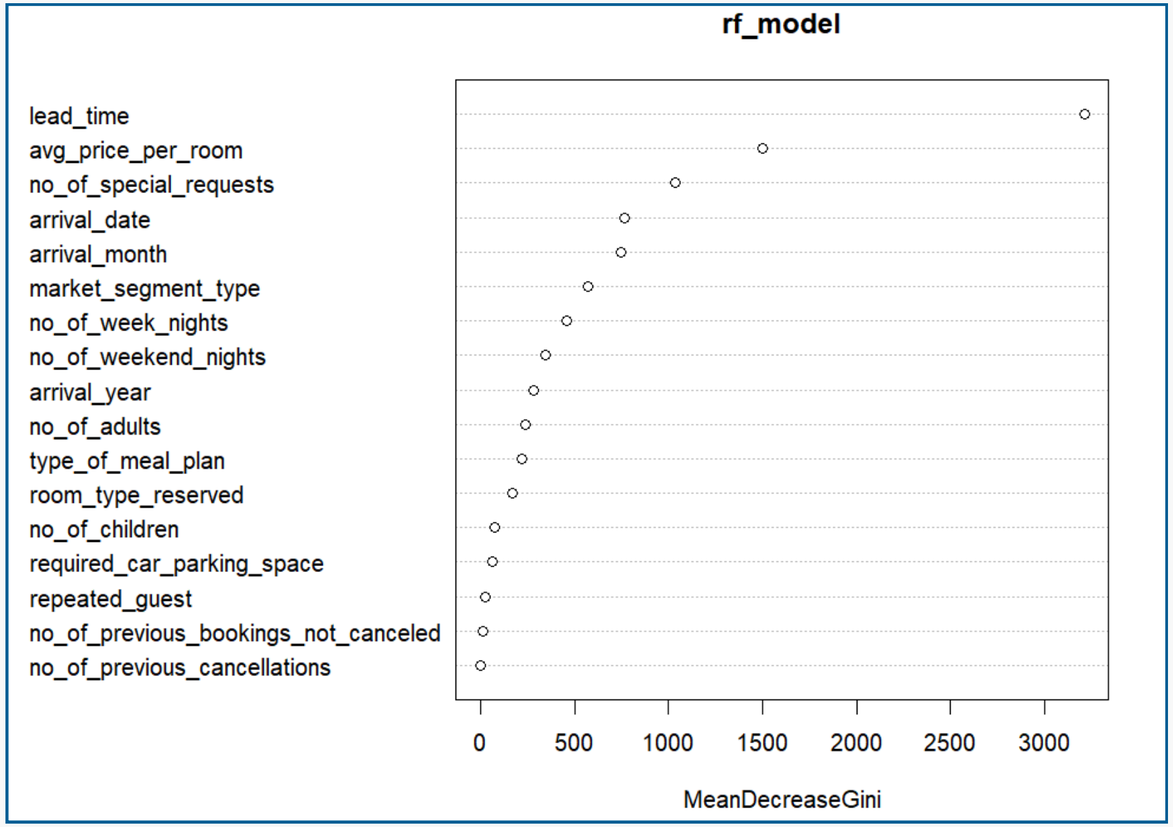
Random Forest Algorithm For Feature Selection

Random Forest untuk pemilihan fitur memanfaatkan kemampuan algoritme untuk mengurutkan fitur berdasarkan tingkat kepentingannya dalam memprediksi variabel target.

Sintaks

```
# FEATURE SELECTION DENGAN RANDOM FOREST
set.seed(123)
train_index <- sample(1:nrow(df), 0.7 * nrow(df))
train_data <- df[train_index, ]
test_data <- df[-train_index, ]
rf_model <- randomForest(booking_status ~ ., data = train_data)
importance <- importance(rf_model)
print(importance)
varImpPlot(rf_model)
```

Output



Method

Feature Selection

▼

RF Feature Importance + Intuitive

Intrepretasi

Mean Decrease Gini (MDG) adalah salah satu metrik yang mengukur seberapa signifikan sebuah fitur dalam menurunkan impuritas (misclassification) di dalam pohon keputusan.

Namun, dalam pemilihan fitur juga diperlukan pertimbangan intuitif untuk menentukan fitur-fitur yang benar-benar berpengaruh terhadap status booking pelanggan hotel. Hal ini penting agar dapat meningkatkan interpretabilitas model dan keefektifan strategi bisnis.

```
selected_variables <- c(
  'lead_time', 'market_segment_type', 'no_of_adults', 'no_of_children',
  'no_of_weekend_nights', 'no_of_week_nights', 'room_type_reserved',
  'arrival_month', 'arrival_date', 'repeated_guest',
  'no_of_previous_cancellations', 'avg_price_per_room','booking_status'
)
df1 <- df[selected_variables]
```

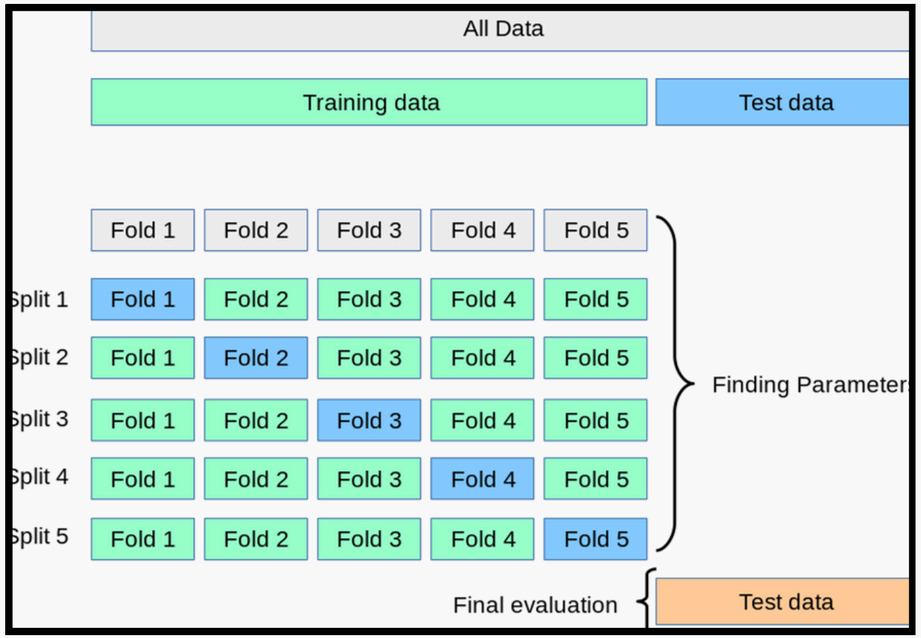
KLASIFIKASI

Decision Tree | Naive Bayes | Regresi Logistik | XGBoost

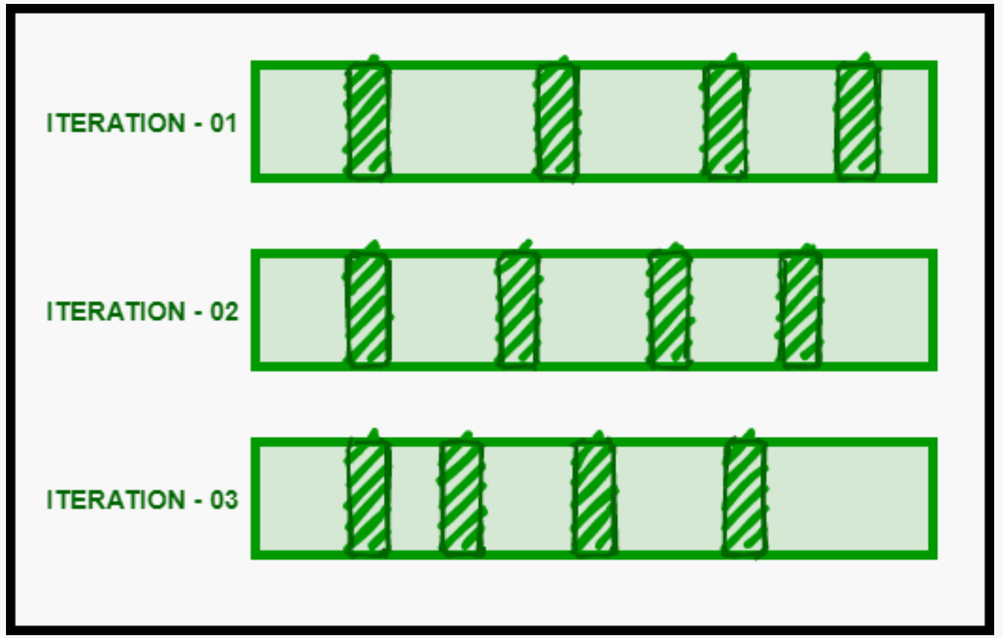
dengan training dan testing menggunakan K-fold dan Holdout

Akurasi | Sensitifitas | Spesifitas | AUC-ROC

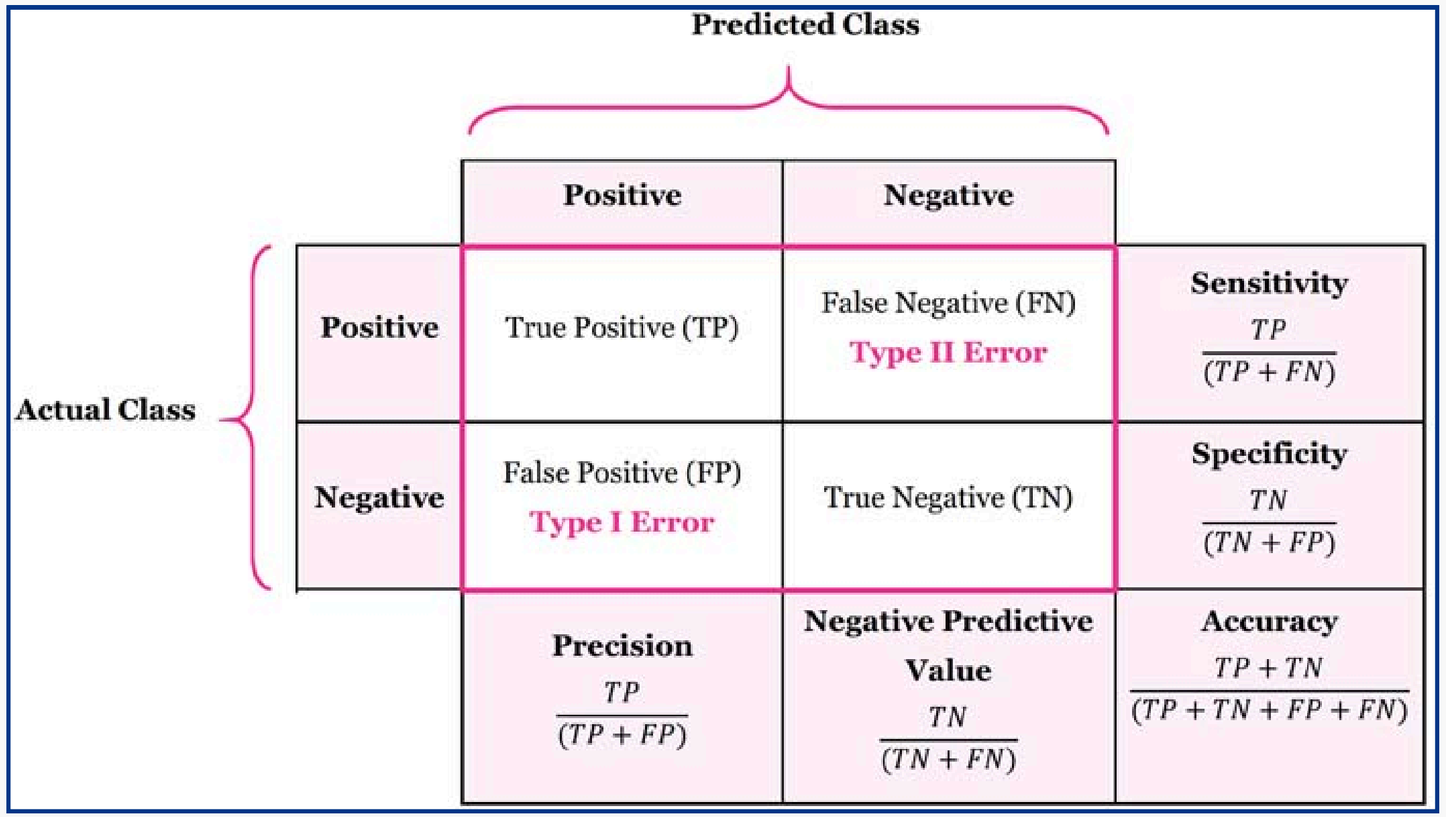
K-Fold



Repeated Holdout

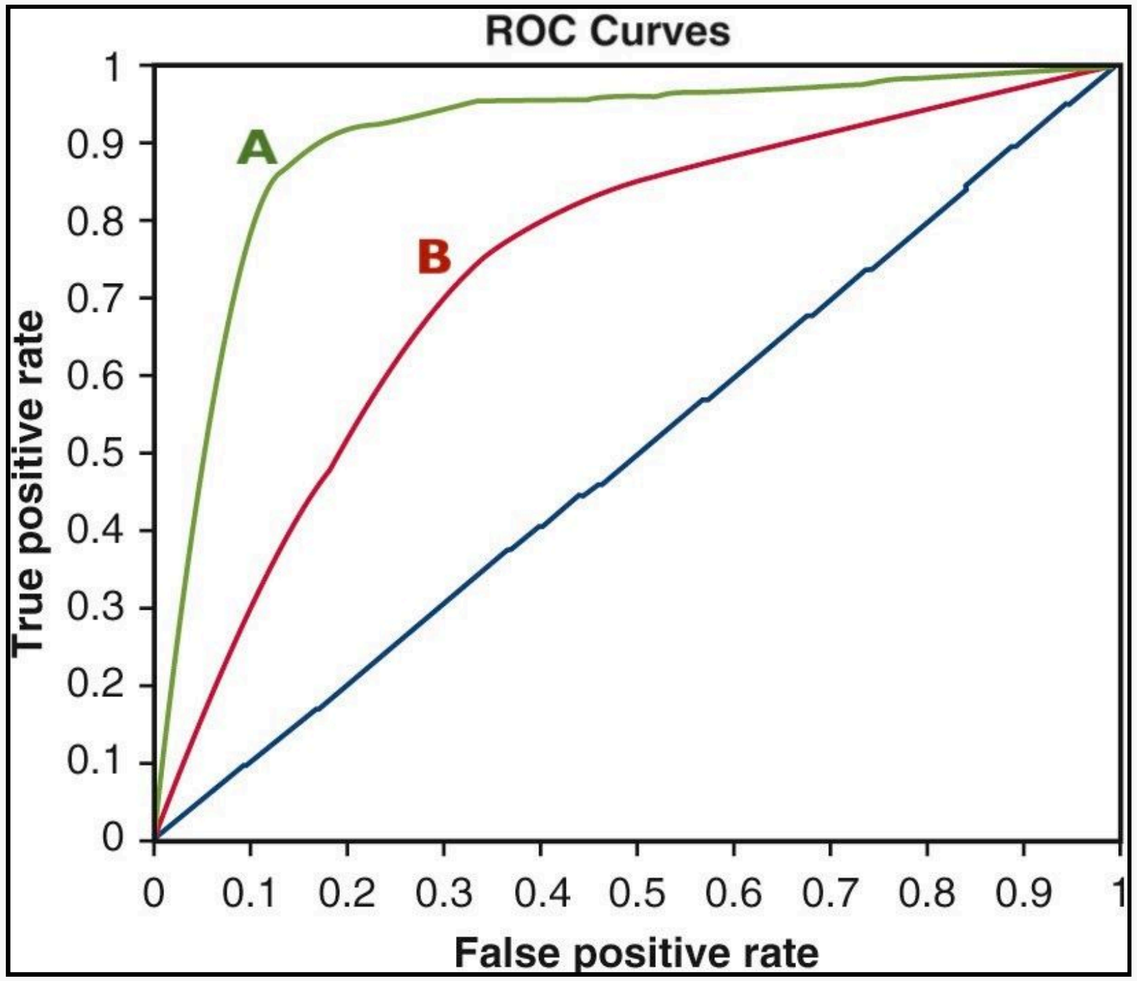


Metrik Pengukuran Performa Model



- Akurasi : proporsi prediksi benar (baik positif maupun negatif) dari keseluruhan prediksi.
- presisi : Mengukur proporsi prediksi positif yang benar-benar positif.
- sensitivitas : Mengukur proporsi kejadian positif yang berhasil diidentifikasi dengan benar oleh model.
- spesifitas : Mengukur proporsi kejadian negatif yang berhasil diidentifikasi dengan benar oleh model.

Kurva AUC-ROC



Kurva ROC (Receiver Operating Characteristic) adalah plot yang menggambarkan kinerja model klasifikasi dengan memplot True Positive Rate (TPR) melawan False Positive Rate (FPR) pada berbagai ambang batas (threshold) klasifikasi. Untuk menggambar kurva ROC, kita perlu menghitung TPR dan FPR pada berbagai threshold.

AUC (Area Under the Curve) adalah luas di bawah kurva ROC, yang memberikan gambaran umum tentang kinerja model.

$$(\text{TPR}) = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}}$$

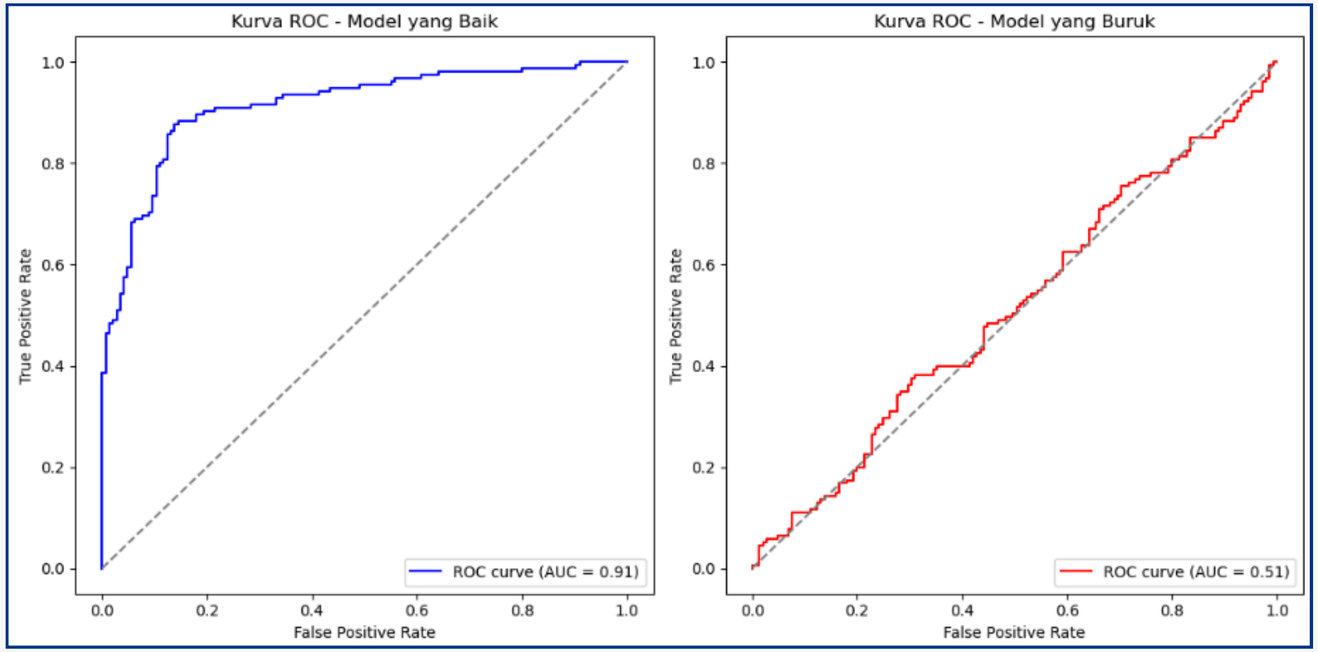
$$\text{Spesifisitas} = \frac{\text{True Negatives (TN)}}{\text{True Negatives (TN)} + \text{False Positives (FP)}}$$

$$\text{FPR} = 1 - \text{Spesifisitas}$$

Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

Intrepretasi Kurva AUC-ROC



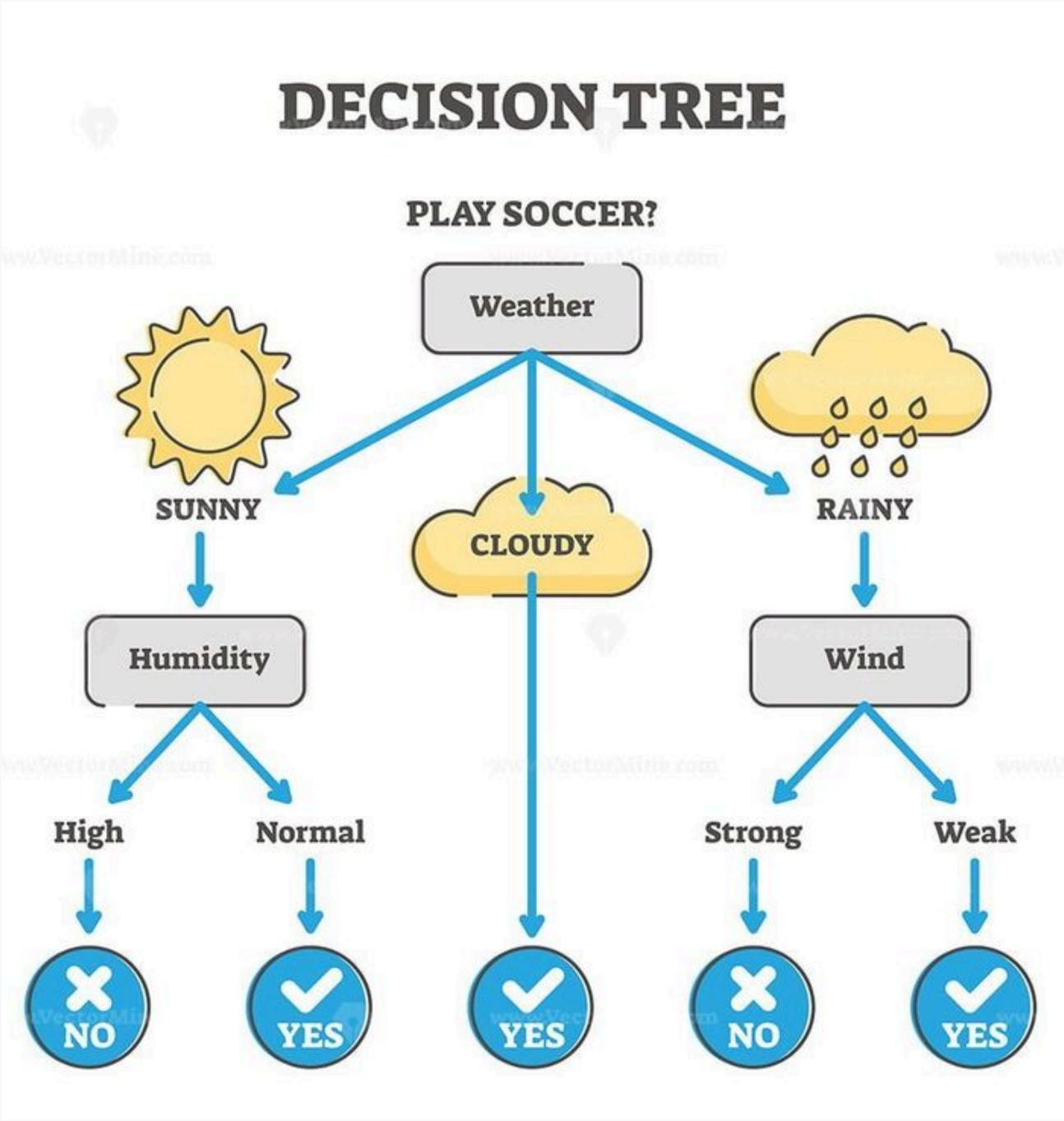
ROC

AUC

AUC	Meaning	Symbol
0.90 - 1.00	excellent classification	
0.80 - 0.90	good classification	
0.70 - 0.80	fair classification	
0.60 - 0.70	poor classification	
< 0.60	failure	

DECISION TREE

Decision Tree membagi data menjadi subset berdasarkan nilai fitur, menciptakan struktur pohon di mana setiap node mewakili fitur, tepi mewakili aturan keputusan, dan node daun menunjukkan label kelas. Decision Tree mudah dipahami, dapat menangkap hubungan non-linear, dan tidak memerlukan penskalaan fitur. Namun, rentan terhadap overfitting dan dapat menjadi bias jika satu kelas mendominasi.



DECISION TREE BASED KFOLD

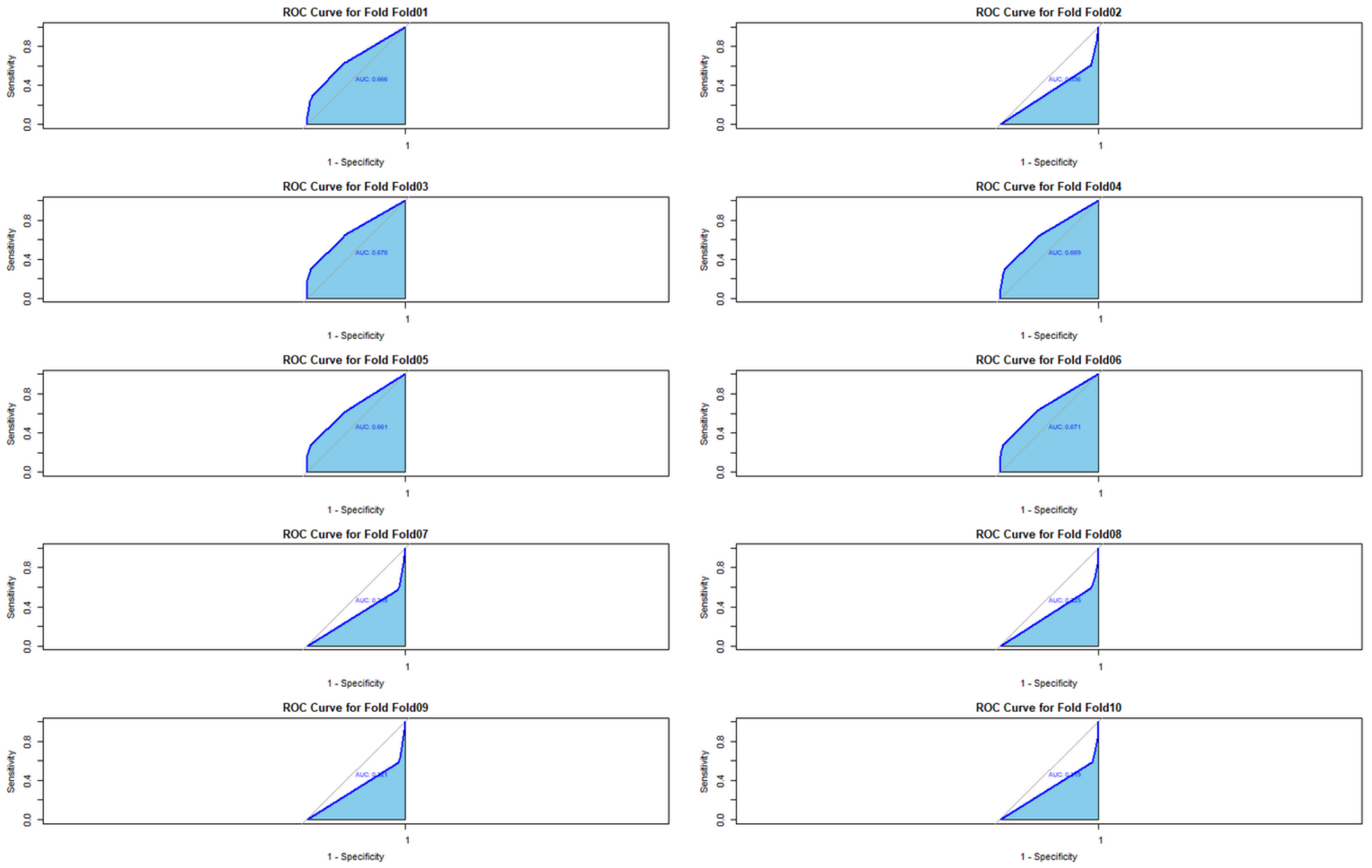
Fold: Fold01
Accuracy: 0.7348401
Sensitivity: 0.2963274
Specificity: 0.9486128
Setting direction: controls > cases
AUC: 0.6658217

Fold: Fold02
Accuracy: 0.7538589
Sensitivity: 0.3910849
Specificity: 0.9307093
Setting direction: controls < cases
AUC: 0.3357629

Fold: Fold03
Accuracy: 0.7428545
Sensitivity: 0.2957351
Specificity: 0.9606396
Setting direction: controls > cases
AUC: 0.6761228

Fold: Fold04
Accuracy: 0.7377079
Sensitivity: 0.297138
Specificity: 0.9523029
Setting direction: controls > cases
AUC: 0.6686238

Fold: Fold05
Accuracy: 0.7362183
Sensitivity: 0.27418
Specificity: 0.9614596
Setting direction: controls > cases
AUC: 0.6606522



Fold: Fold06
Accuracy: 0.7424869
Sensitivity: 0.2758137
Specificity: 0.9697964
Setting direction: controls > cases
AUC: 0.6706362

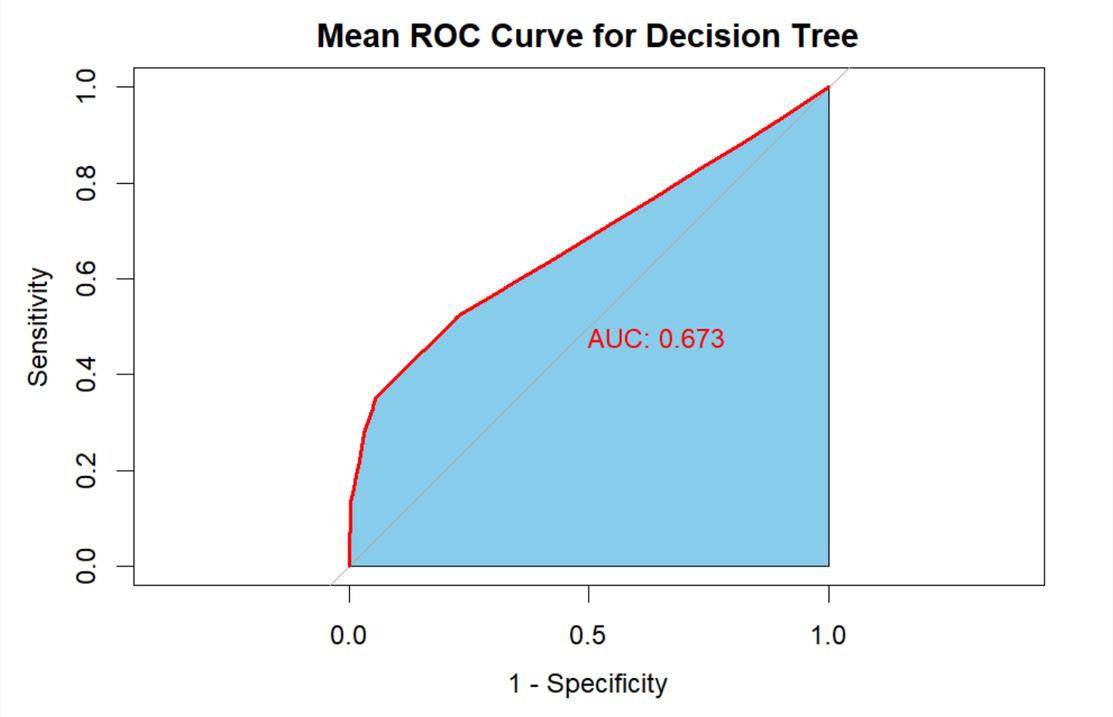
Fold: Fold07
Accuracy: 0.764268
Sensitivity: 0.4208754
Specificity: 0.9315293
Setting direction: controls < cases
AUC: 0.3175872

Fold: Fold08
Accuracy: 0.757534
Sensitivity: 0.3989347
Specificity: 0.9323493
Setting direction: controls < cases
AUC: 0.3254073

Fold: Fold09
Accuracy: 0.7627975
Sensitivity: 0.4197531
Specificity: 0.9298893
Setting direction: controls < cases
AUC: 0.3211199

Fold: Fold10
Accuracy: 0.7651599
Sensitivity: 0.4107093
Specificity: 0.9379527
Setting direction: controls < cases
AUC: 0.3190241

MEAN AUC BASED KFOLD



DECISION TREE BASED HOLDOUT

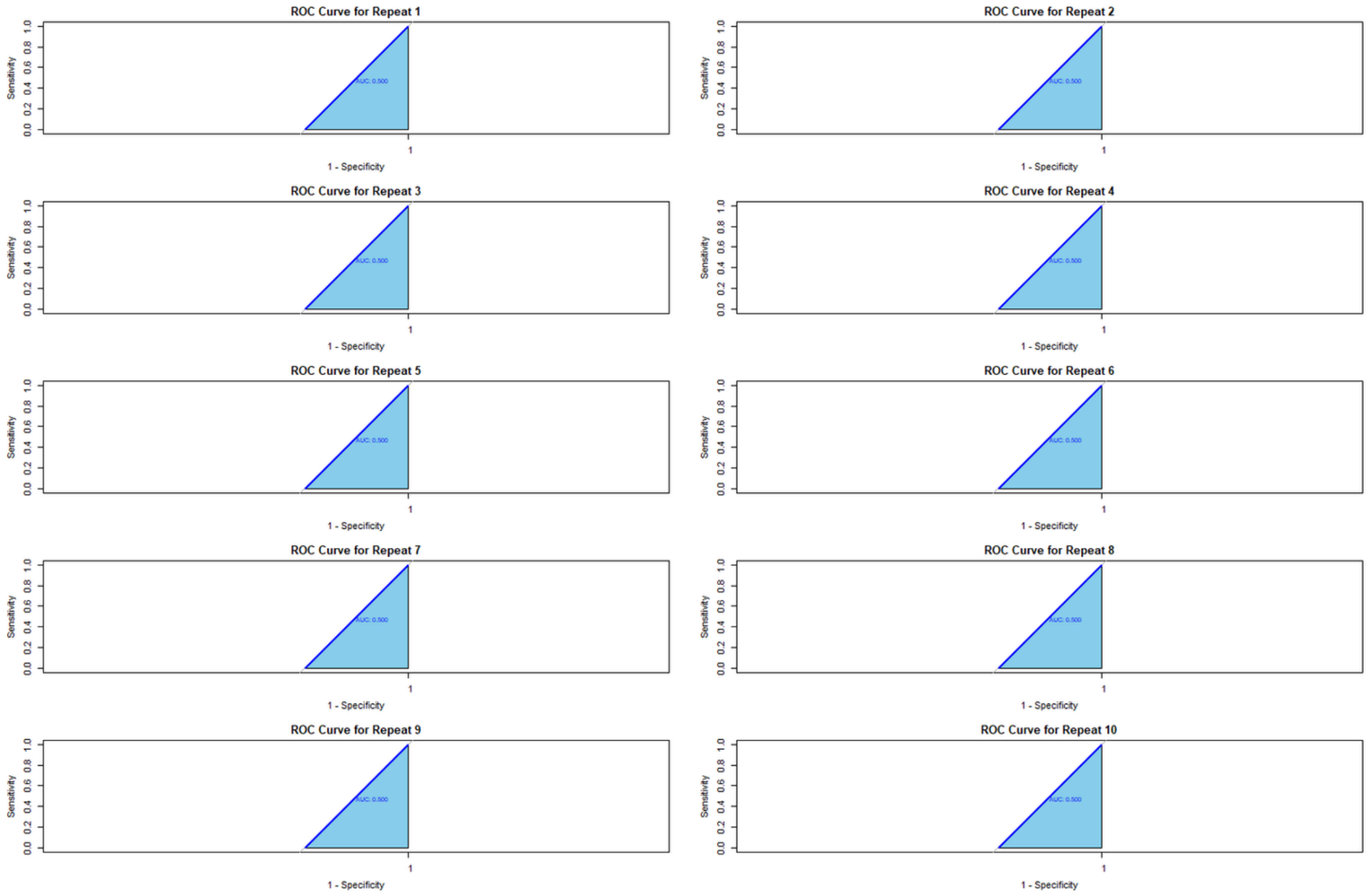
Repeat: 1
Accuracy: 0.6698521
Sensitivity: 0
Specificity: 1
Setting direction: controls < cases
AUC: 0.5

Repeat: 2
Accuracy: 0.678765
Sensitivity: 0
Specificity: 1
Setting direction: controls < cases
AUC: 0.5

Repeat: 3
Accuracy: 0.6782137
Sensitivity: 0
Specificity: 1
Setting direction: controls < cases
AUC: 0.5

Repeat: 4
Accuracy: 0.6734356
Sensitivity: 0
Specificity: 1
Setting direction: controls < cases
AUC: 0.5

Repeat: 5
Accuracy: 0.6703115
Sensitivity: 0
Specificity: 1
Setting direction: controls < cases
AUC: 0.5



Repeat: 6
Accuracy: 0.6684738
Sensitivity: 0
Specificity: 1
Setting direction: controls < cases
AUC: 0.5

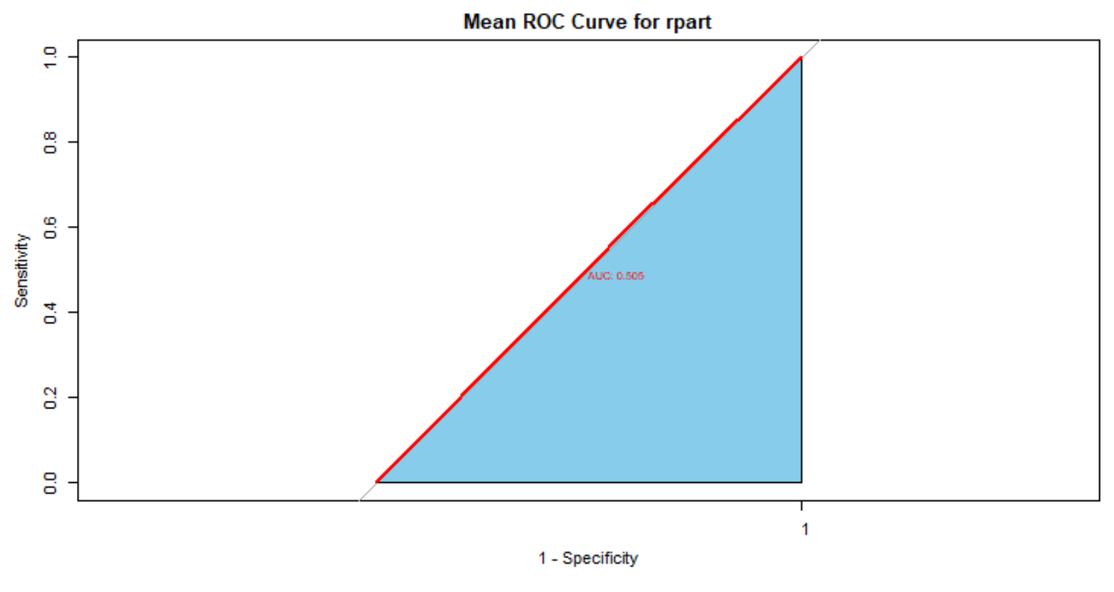
Repeat: 7
Accuracy: 0.6668198
Sensitivity: 0
Specificity: 1
Setting direction: controls < cases
AUC: 0.5

Repeat: 8
Accuracy: 0.6688413
Sensitivity: 0
Specificity: 1
Setting direction: controls < cases
AUC: 0.5

Repeat: 9
Accuracy: 0.6684738
Sensitivity: 0
Specificity: 1
Setting direction: controls < cases
AUC: 0.5

Repeat: 10
Accuracy: 0.672333
Sensitivity: 0
Specificity: 1
Setting direction: controls < cases
AUC: 0.5

MEAN AUC BASED HOLDOUT




```
AUC values for each repeat:
 [1] 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5

Mean AUC: 0.5
```

NAIVE BAYES

Naive Bayes adalah algoritma klasifikasi probabilistik yang cepat dan efisien berdasarkan teorema Bayes, dengan mengasumsikan independensi fitur yang diberikan label kelas. Algoritma ini unggul dengan dataset besar dan biasanya digunakan dalam deteksi spam, analisis sentimen, dan klasifikasi dokumen. Namun, asumsi independensi fitur dapat menyebabkan ketidakakuratan jika variabel berkorelasi, dan mungkin mengalami kesulitan dengan dataset yang kecil atau tidak representatif.

NAIVE BAYES BASED KFOLD

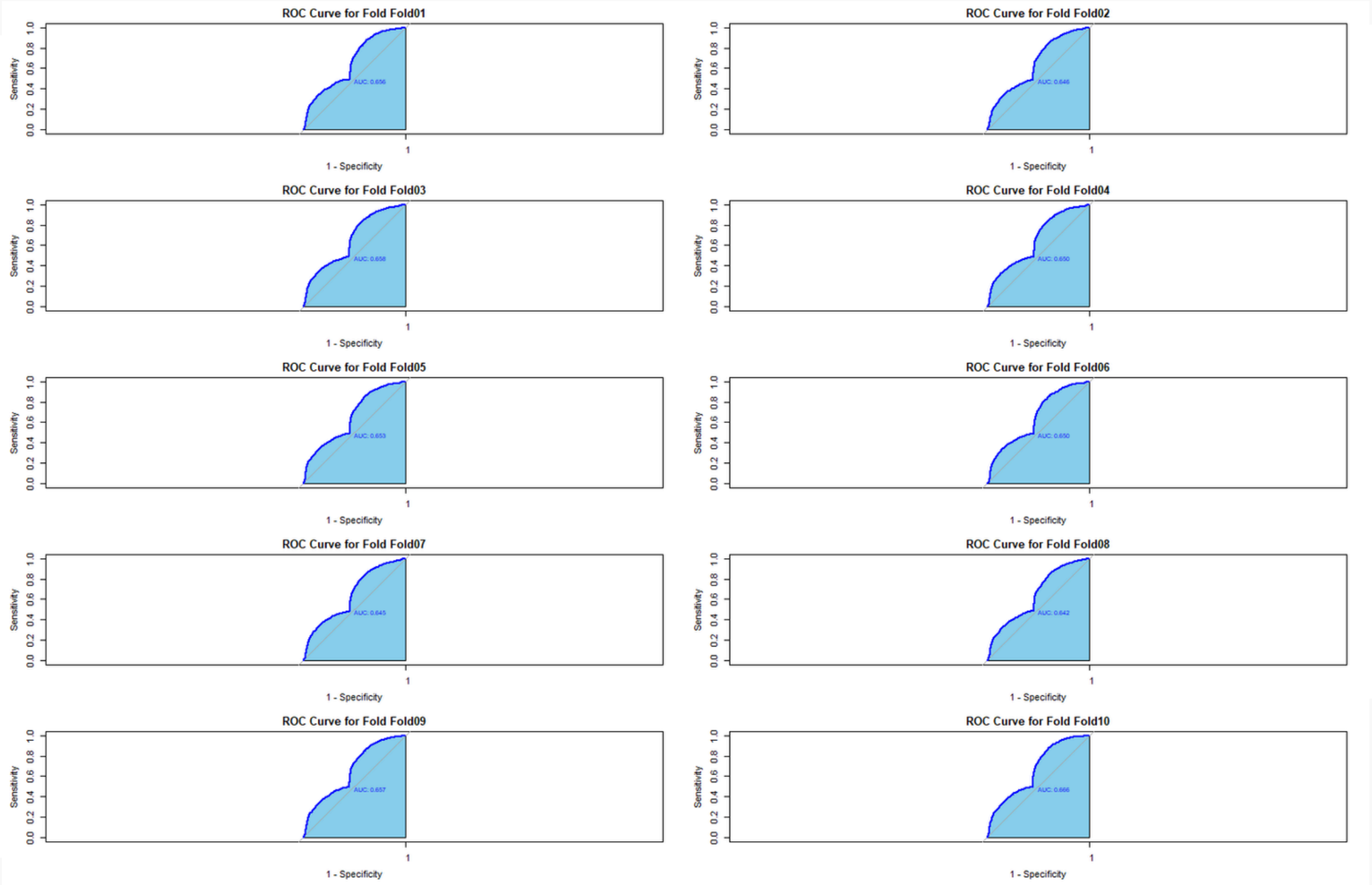
Fold: Fold01
Accuracy: 0.5512679
Sensitivity: 0.5786375
Specificity: 0.5379254
Setting direction: controls > cases
AUC: 0.6558237

Fold: Fold02
Accuracy: 0.561742
Sensitivity: 0.6013457
Specificity: 0.5424354
Setting direction: controls > cases
AUC: 0.6463878

Fold: Fold03
Accuracy: 0.5774745
Sensitivity: 0.6544613
Specificity: 0.5399754
Setting direction: controls > cases
AUC: 0.657871

Fold: Fold04
Accuracy: 0.5548663
Sensitivity: 0.5904882
Specificity: 0.5375154
Setting direction: controls > cases
AUC: 0.6500849

Fold: Fold05
Accuracy: 0.557194
Sensitivity: 0.5891505
Specificity: 0.5416154
Setting direction: controls > cases
AUC: 0.6525501



Fold: Fold06
Accuracy: 0.5550041
Sensitivity: 0.5934343
Specificity: 0.5362854
Setting direction: controls > cases
AUC: 0.6501425

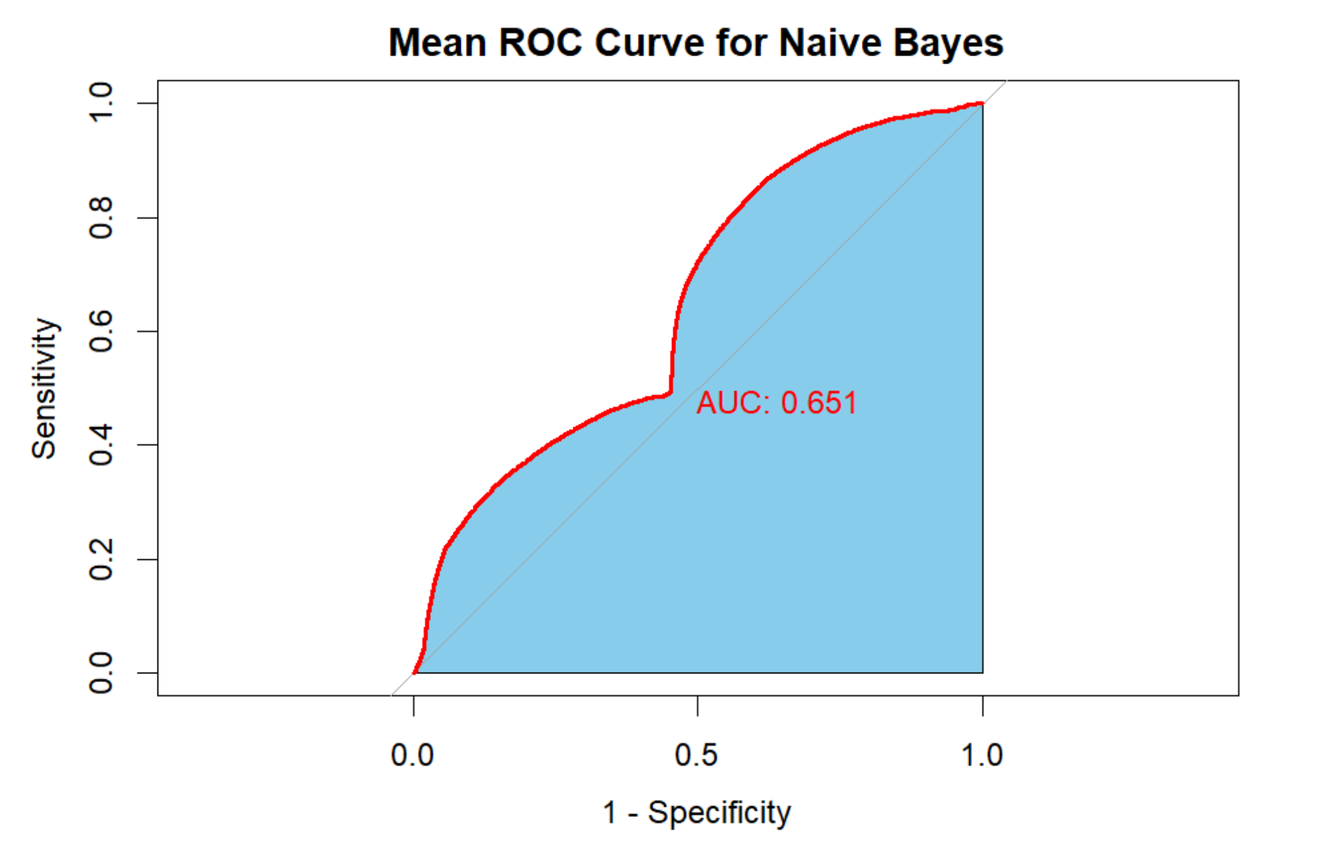
Fold: Fold07
Accuracy: 0.5515578
Sensitivity: 0.5736532
Specificity: 0.5407954
Setting direction: controls > cases
AUC: 0.6447959

Fold: Fold08
Accuracy: 0.5538864
Sensitivity: 0.5933558
Specificity: 0.5346453
Setting direction: controls > cases
AUC: 0.6415557

Fold: Fold09
Accuracy: 0.5454921
Sensitivity: 0.5395623
Specificity: 0.5483805
Setting direction: controls > cases
AUC: 0.6569357

Fold: Fold10
Accuracy: 0.5595369
Sensitivity: 0.5769554
Specificity: 0.5510455
Setting direction: controls > cases
AUC: 0.6655326

MEAN AUC BASED KFOLD



NAIVE BAYES BASED HOLDOUT

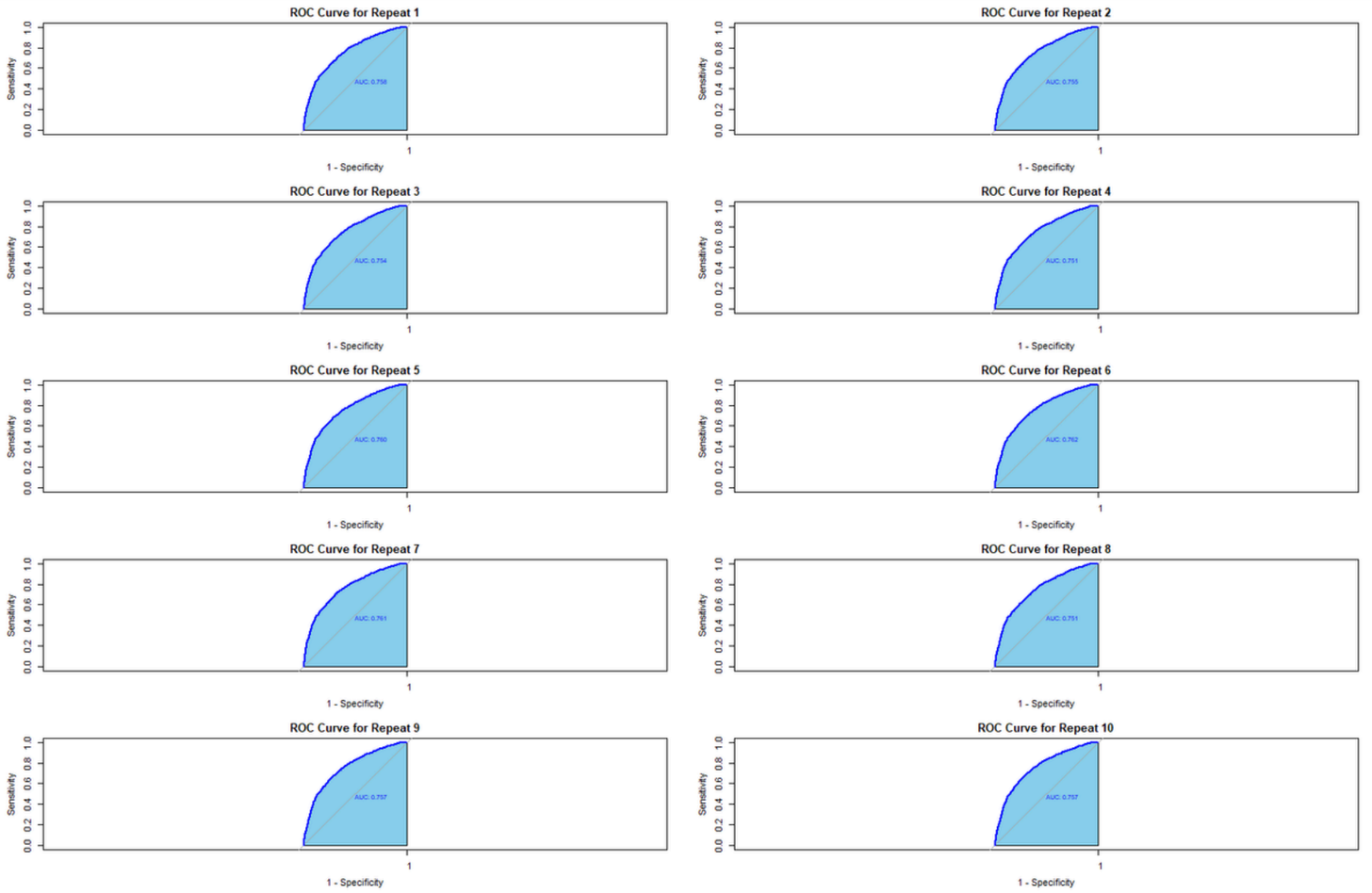
Repeat: 1
Accuracy: 0.7285675
Sensitivity: 0.5527414
Specificity: 0.8152263
Setting direction: controls > cases
AUC: 0.7575104

Repeat: 2
Accuracy: 0.7335294
Sensitivity: 0.5445658
Specificity: 0.8222822
Setting direction: controls > cases
AUC: 0.754942

Repeat: 3
Accuracy: 0.7259028
Sensitivity: 0.5627496
Specificity: 0.8034431
Setting direction: controls > cases
AUC: 0.7538889

Repeat: 4
Accuracy: 0.7279243
Sensitivity: 0.5354571
Specificity: 0.8234566
Setting direction: controls > cases
AUC: 0.7506367

Repeat: 5
Accuracy: 0.7382156
Sensitivity: 0.5596542
Specificity: 0.8217995
Setting direction: controls > cases
AUC: 0.7599475



Repeat: 6
Accuracy: 0.7368373
Sensitivity: 0.5403135
Specificity: 0.8328546
Setting direction: controls > cases
AUC: 0.7615595

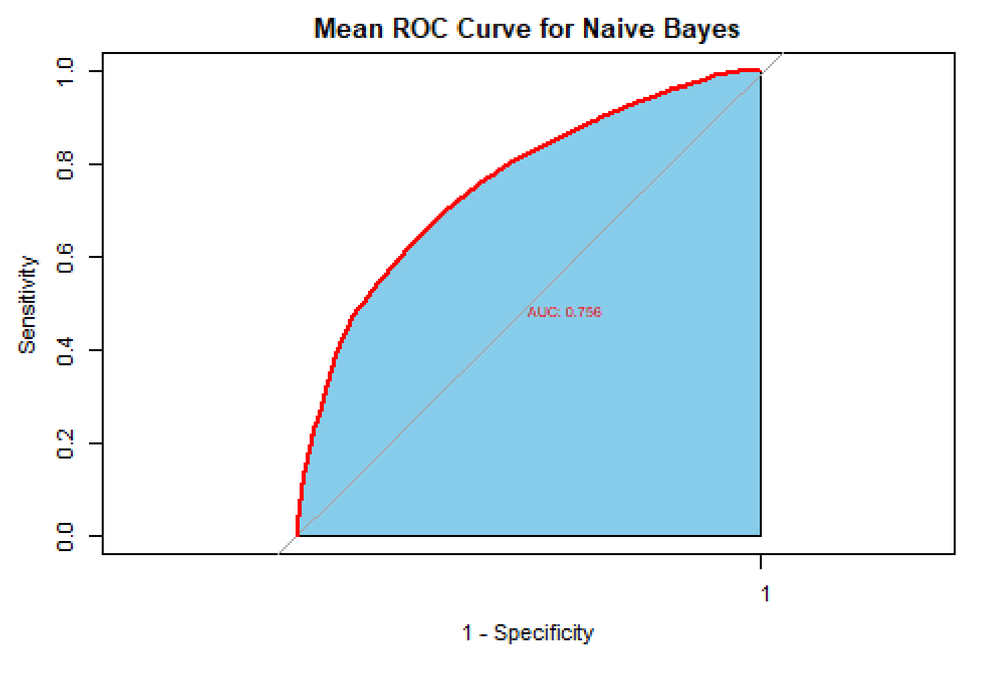
Repeat: 7
Accuracy: 0.7311403
Sensitivity: 0.5672727
Specificity: 0.8113027
Setting direction: controls > cases
AUC: 0.7611583

Repeat: 8
Accuracy: 0.724892
Sensitivity: 0.5391256
Specificity: 0.8163741
Setting direction: controls > cases
AUC: 0.7506042

Repeat: 9
Accuracy: 0.7356427
Sensitivity: 0.5262862
Specificity: 0.8372918
Setting direction: controls > cases
AUC: 0.7565045

Repeat: 10
Accuracy: 0.729762
Sensitivity: 0.5493348
Specificity: 0.819244
Setting direction: controls > cases
AUC: 0.7566597

MEAN AUC BASED HOLDOUT




```
AUC values for each repeat:
 [1] 0.7575104 0.7549420 0.7538889 0.7506367 0.7599475 0.7615595
 [7] 0.7611583 0.7506042 0.7565045 0.7566597

Mean AUC: 0.7563412
```

REGRESI LOGISTIK

Regresi logistik adalah algoritma klasifikasi yang memodelkan probabilitas suatu data termasuk dalam kategori tertentu menggunakan fungsi logistik sigmoid. Kelebihannya meliputi kesederhanaan, kecepatan, dan kemampuan menghasilkan probabilitas yang informatif. Namun, regresi logistik kurang efektif untuk klasifikasi non-linear tanpa fitur tambahan dan performanya dapat menurun dengan multikolinearitas atau data yang tidak terpisah secara linear.

REGRESI LOGISTIK BASED KFOLD

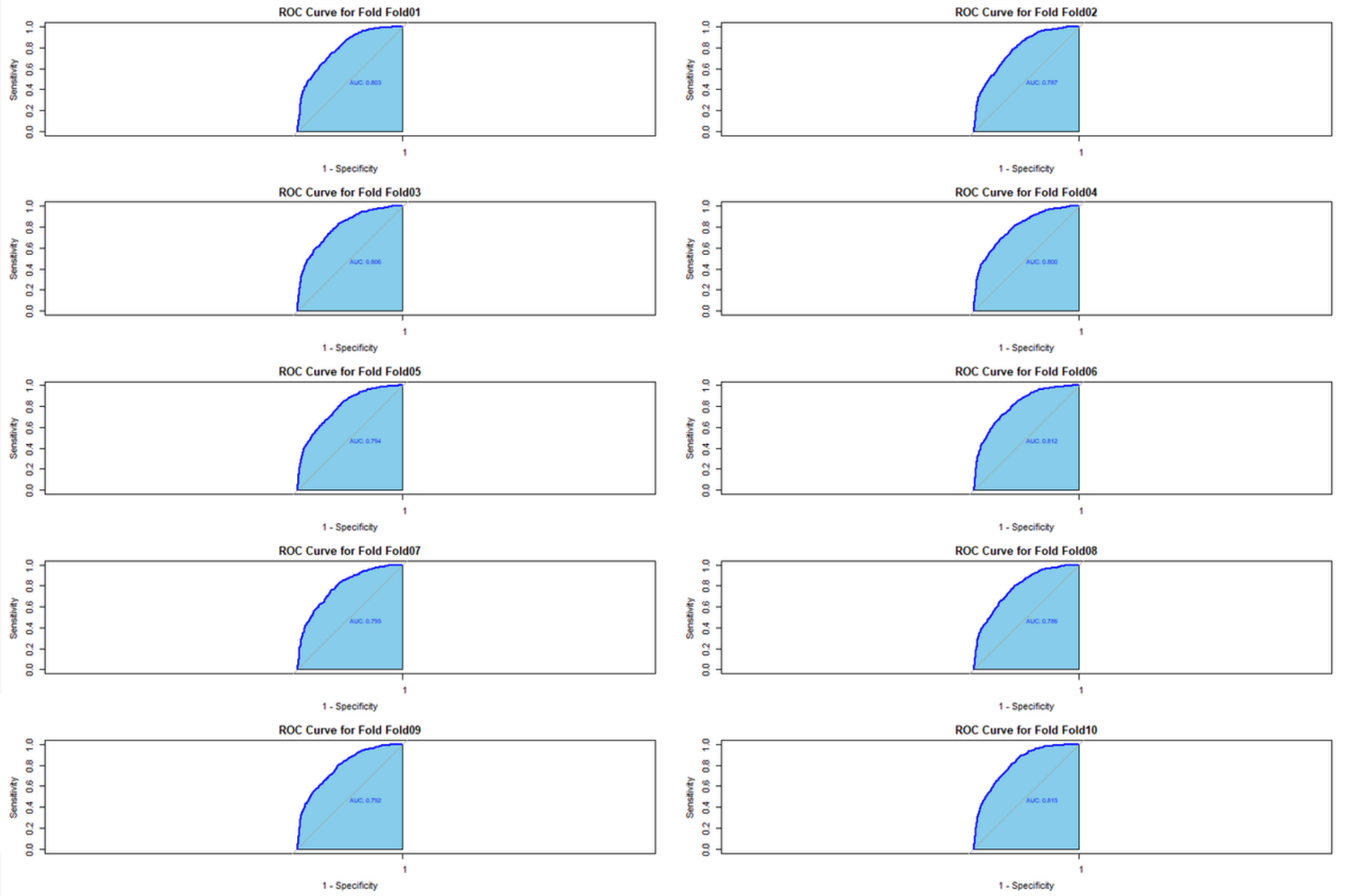
Fold: Fold01
Accuracy: 0.7538589
Sensitivity: 0.4970563
Specificity: 0.8790488
Setting direction: controls > cases
AUC: 0.8028944

Fold: Fold02
Accuracy: 0.7439361
Sensitivity: 0.4474348
Specificity: 0.8884789
Setting direction: controls > cases
AUC: 0.7874912

Fold: Fold03
Accuracy: 0.762338
Sensitivity: 0.5042088
Specificity: 0.8880689
Setting direction: controls > cases
AUC: 0.8057578

Fold: Fold04
Accuracy: 0.7590295
Sensitivity: 0.4890572
Specificity: 0.8905289
Setting direction: controls > cases
AUC: 0.800251

Fold: Fold05
Accuracy: 0.7555127
Sensitivity: 0.470984
Specificity: 0.8942189
Setting direction: controls > cases
AUC: 0.7944414



Fold: Fold06
Accuracy: 0.7612352
Sensitivity: 0.5143098
Specificity: 0.8815088
Setting direction: controls > cases
AUC: 0.8122285

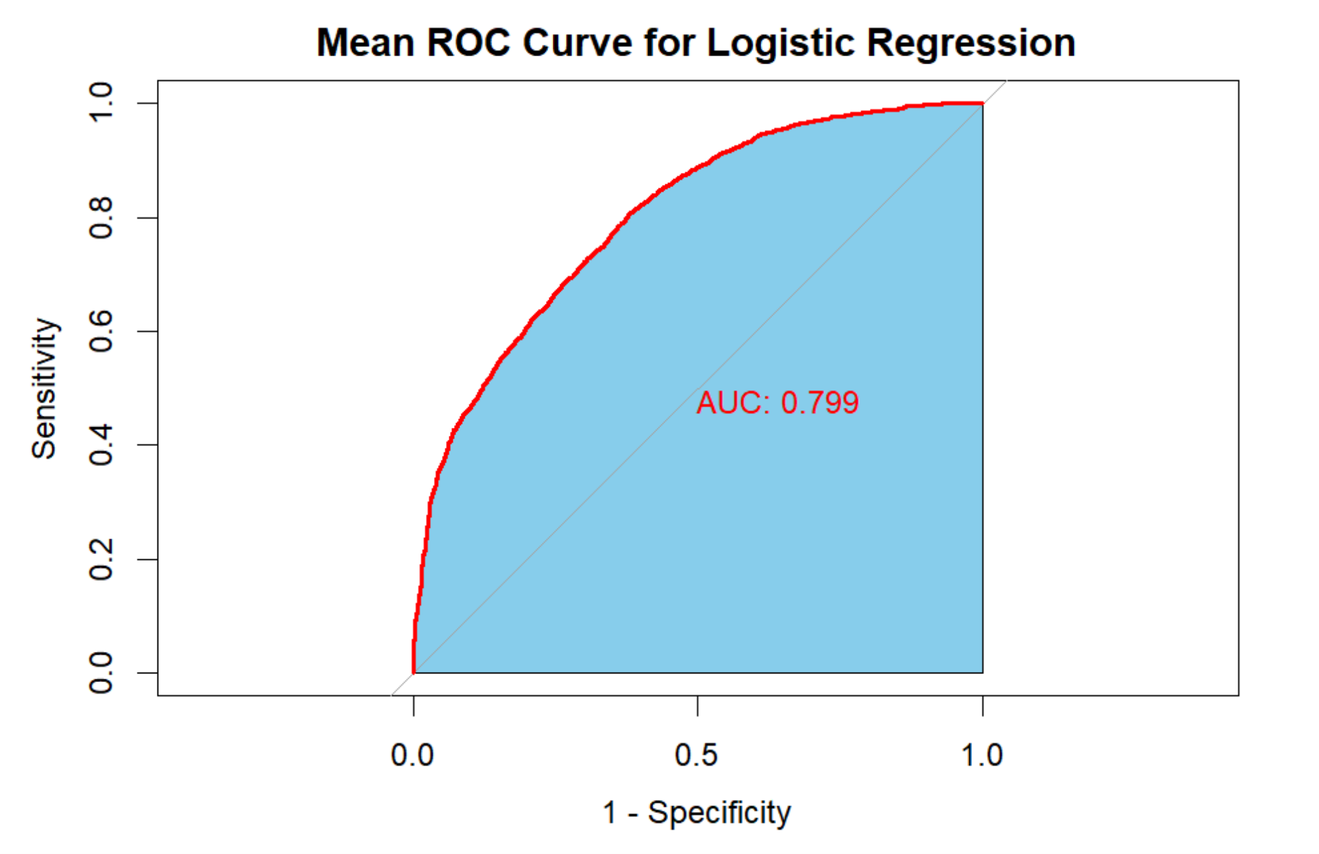
Fold: Fold07
Accuracy: 0.7524125
Sensitivity: 0.479798
Specificity: 0.8851989
Setting direction: controls > cases
AUC: 0.7951172

Fold: Fold08
Accuracy: 0.7422822
Sensitivity: 0.4423886
Specificity: 0.8884789
Setting direction: controls > cases
AUC: 0.7859015

Fold: Fold09
Accuracy: 0.755721
Sensitivity: 0.4823232
Specificity: 0.8888889
Setting direction: controls > cases
AUC: 0.7920599

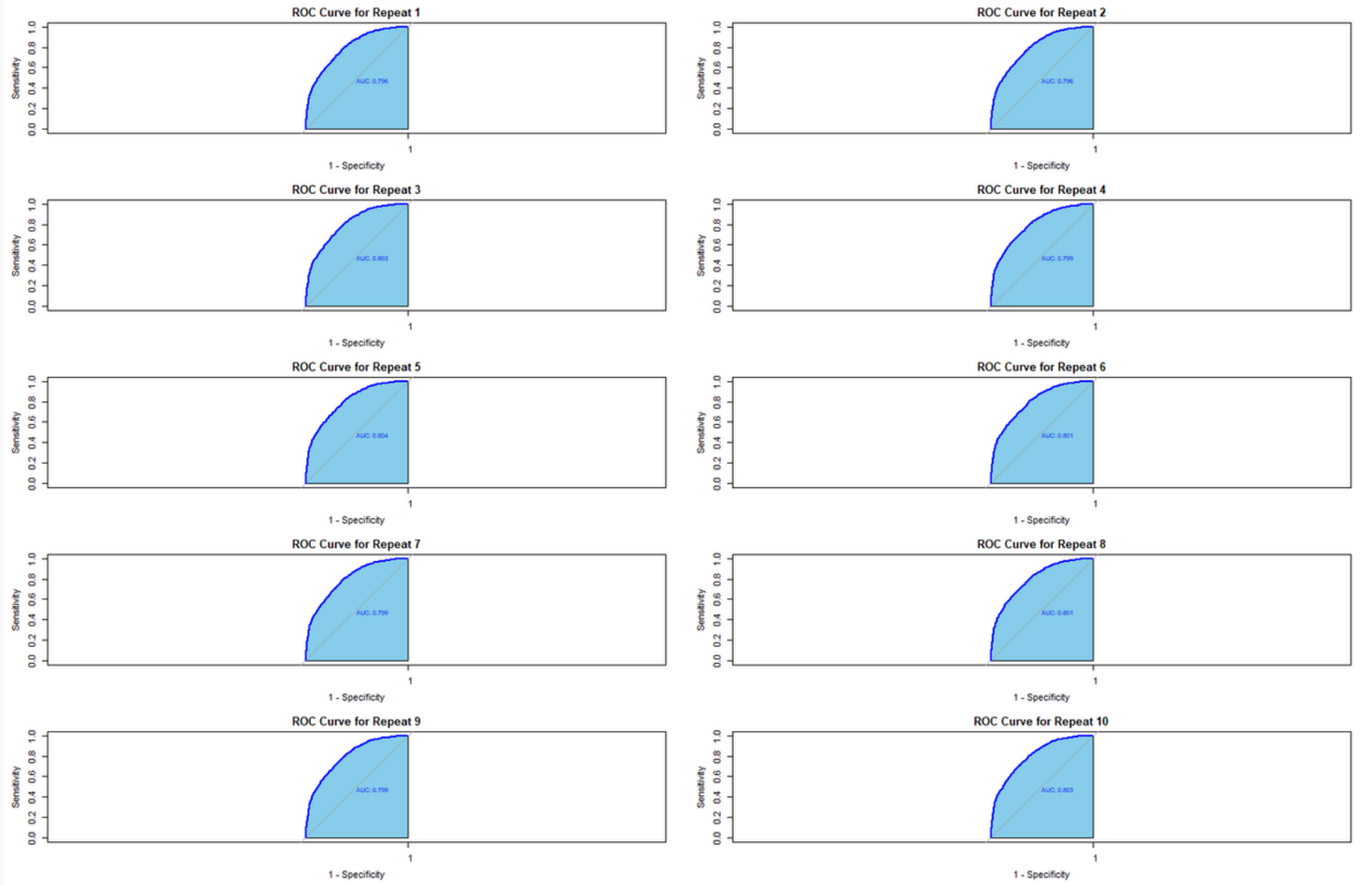
Fold: Fold10
Accuracy: 0.7599228
Sensitivity: 0.4861228
Specificity: 0.8933989
Setting direction: controls > cases
AUC: 0.8146923

MEAN AUC BASED KFOLD



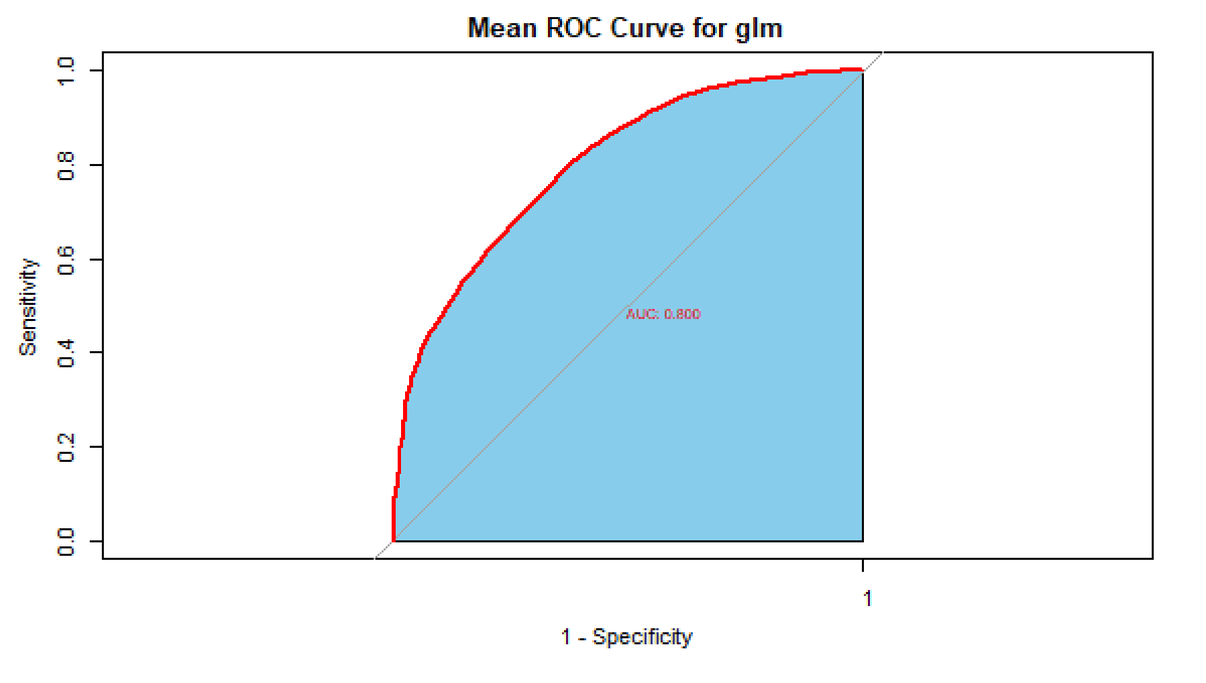
REGRESI LOGISTIK BASED HOLDOUT

```
Repeat: 1
Accuracy: 0.7532849
Sensitivity: 0.4653493
Specificity: 0.8951989
Setting direction: controls > cases
AUC: 0.7960577

Repeat: 2
Accuracy: 0.7561334
Sensitivity: 0.4727582
Specificity: 0.8918331
Setting direction: controls > cases
AUC: 0.7963061

Repeat: 3
Accuracy: 0.7584306
Sensitivity: 0.4814294
Specificity: 0.8927548
Setting direction: controls > cases
AUC: 0.8027635

Repeat: 4
Accuracy: 0.7537444
Sensitivity: 0.4787645
Specificity: 0.8911396
Setting direction: controls > cases
AUC: 0.7986372

Repeat: 5
Accuracy: 0.762933
Sensitivity: 0.4981476
Specificity: 0.8889341
Setting direction: controls > cases
AUC: 0.8043341
```

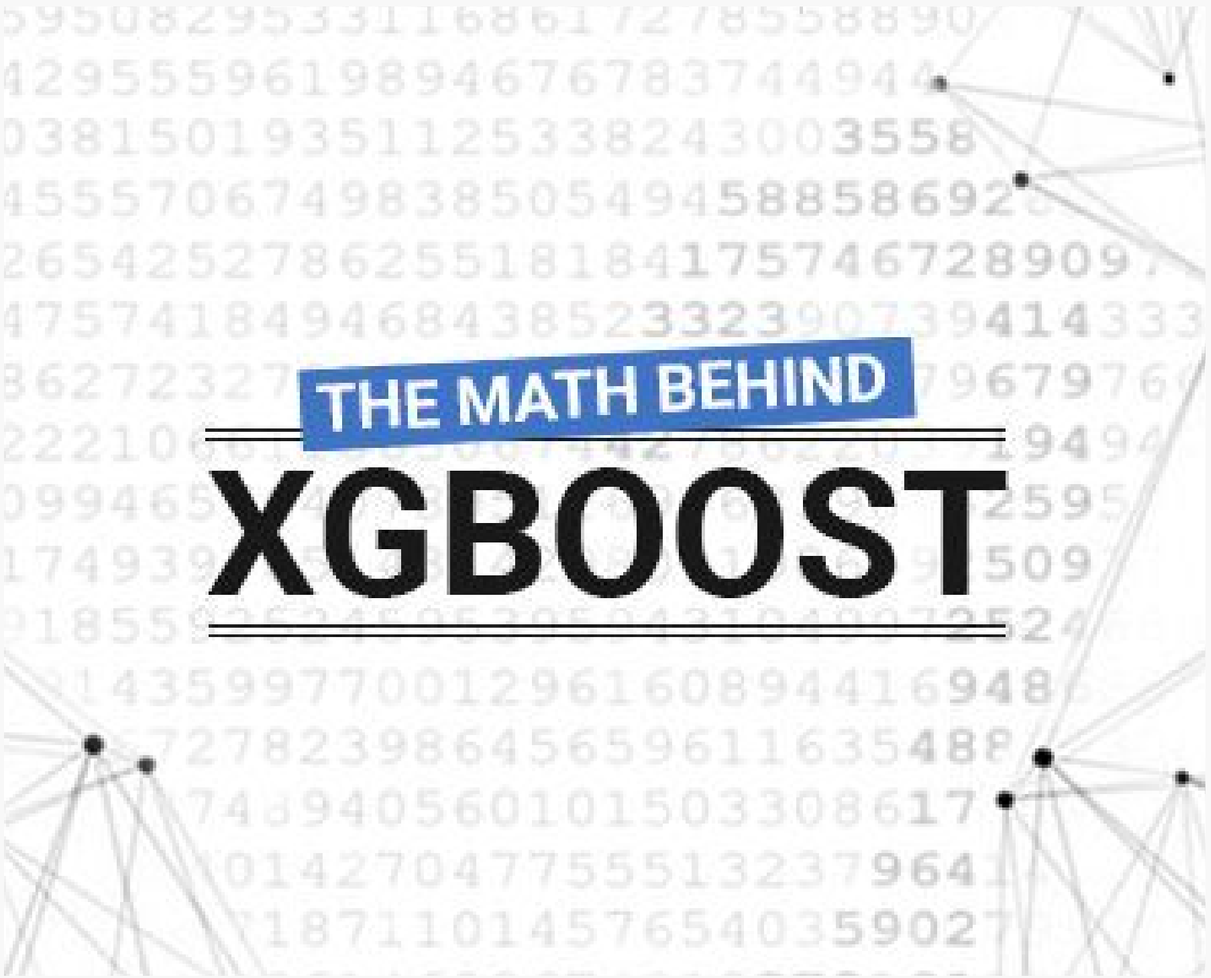



```
Repeat: 6
Accuracy: 0.7587981
Sensitivity: 0.4941011
Specificity: 0.8874778
Setting direction: controls > cases
AUC: 0.8007389

Repeat: 7
Accuracy: 0.7547551
Sensitivity: 0.4816667
Specificity: 0.8897432
Setting direction: controls > cases
AUC: 0.7988434

Repeat: 8
Accuracy: 0.7586144
Sensitivity: 0.4805486
Specificity: 0.894528
Setting direction: controls > cases
AUC: 0.8007902

Repeat: 9
Accuracy: 0.7559496
Sensitivity: 0.488359
Specificity: 0.8863077
Setting direction: controls > cases
AUC: 0.7994242

Repeat: 10
Accuracy: 0.756501
Sensitivity: 0.4820584
Specificity: 0.8918771
Setting direction: controls > cases
AUC: 0.8025889
```

MEAN AUC BASED HOLDOUT




```
AUC values for each repeat:
 [1] 0.7960577 0.7963061 0.8027635 0.7986372 0.8043341 0.8007389
 [7] 0.7988434 0.8007902 0.7994242 0.8025889

Mean AUC: 0.8000484
```



XGBOOST

XGBoost (eXtreme Gradient Boosting) menggunakan pendekatan ensemble dengan decision tree berurutan. Algoritma ini mampu mengatasi data kompleks dan besar dengan performa tinggi. XGBoost juga efektif dalam menangani data tidak seimbang. Namun, pengaturan parameter yang kompleks dan waktu komputasi untuk tuning model bisa menjadi tantangan. Meskipun demikian, dengan parameter yang tepat, XGBoost menghasilkan model yang kuat dan handal untuk berbagai masalah klasifikasi.

XGBOOST BASED KFOLD

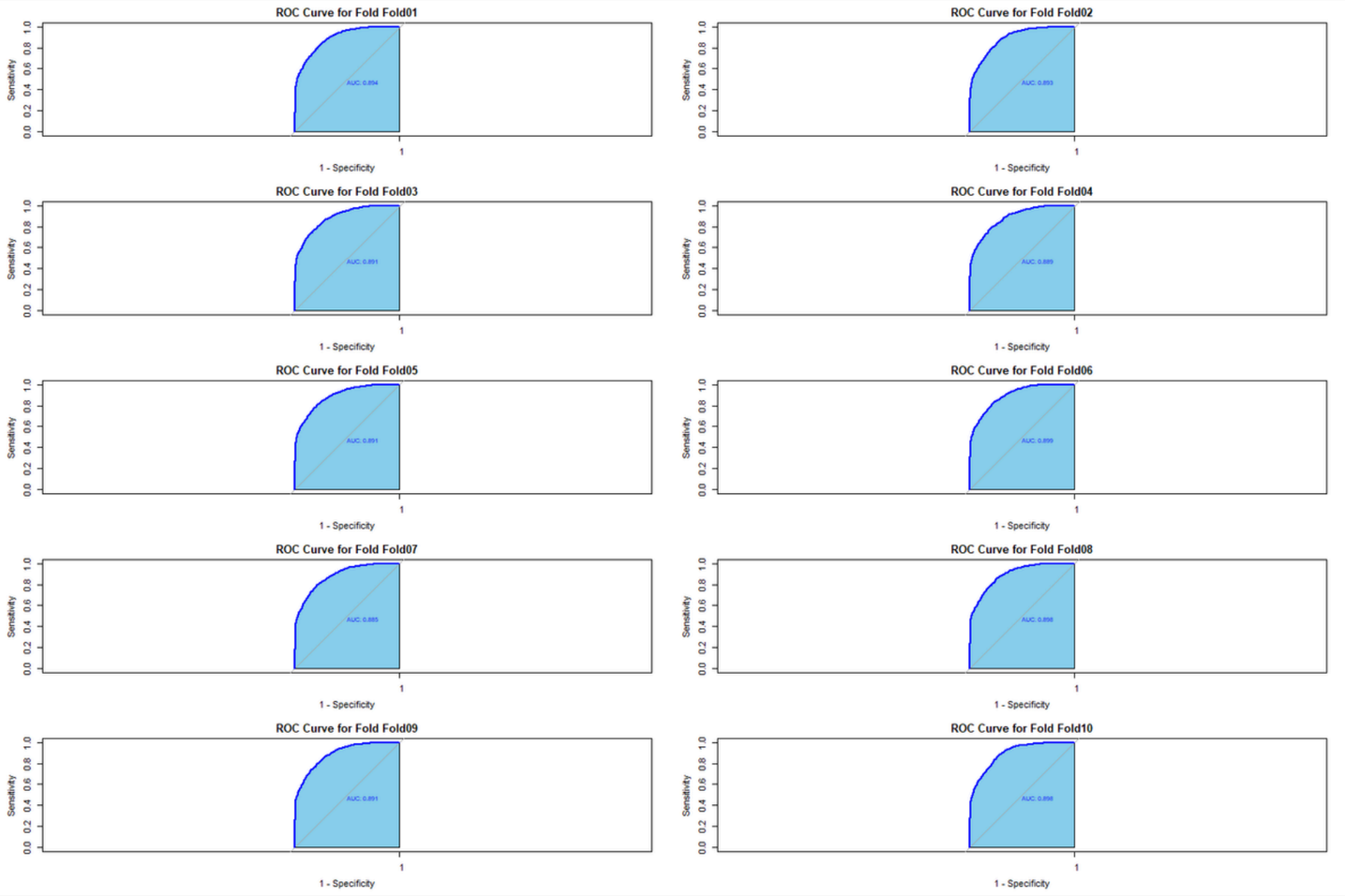
Fold: Fold01
Accuracy: 0.8244212
Sensitivity: 0.569386
Specificity: 0.9487495
Setting direction: controls > cases
AUC: 0.8936776

Fold: Fold02
Accuracy: 0.8233186
Sensitivity: 0.5618167
Specificity: 0.9507995
Setting direction: controls > cases
AUC: 0.8927331

Fold: Fold03
Accuracy: 0.8249242
Sensitivity: 0.5883838
Specificity: 0.9401394
Setting direction: controls > cases
AUC: 0.8910345

Fold: Fold04
Accuracy: 0.8290598
Sensitivity: 0.5707071
Specificity: 0.9548995
Setting direction: controls > cases
AUC: 0.889234

Fold: Fold05
Accuracy: 0.8299338
Sensitivity: 0.5710681
Specificity: 0.9561296
Setting direction: controls > cases
AUC: 0.8914937



Fold: Fold06
Accuracy: 0.8342983
Sensitivity: 0.5850168
Specificity: 0.9557196
Setting direction: controls > cases
AUC: 0.8989906

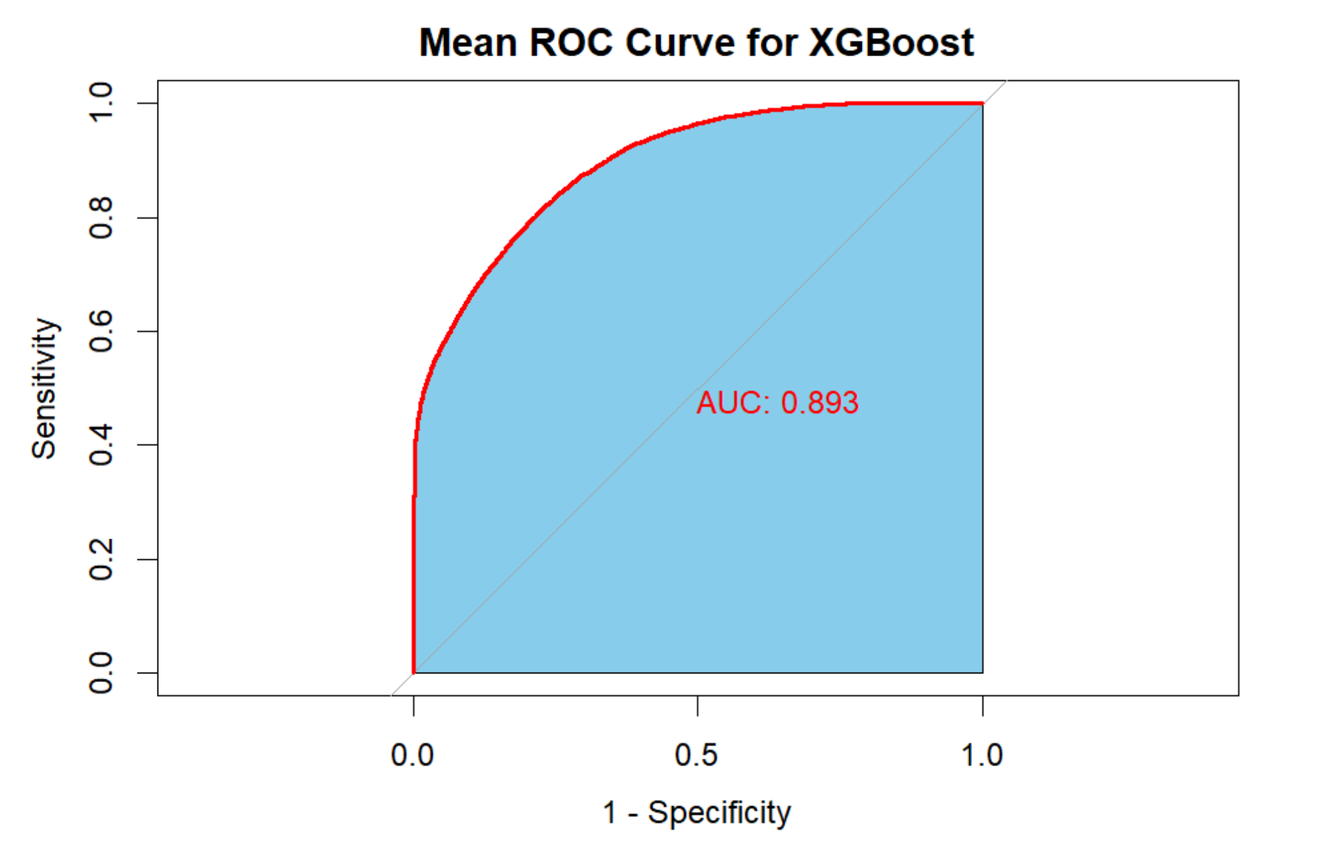
Fold: Fold07
Accuracy: 0.8158258
Sensitivity: 0.5673401
Specificity: 0.9368594
Setting direction: controls > cases
AUC: 0.8845338

Fold: Fold08
Accuracy: 0.8291069
Sensitivity: 0.5828427
Specificity: 0.9491595
Setting direction: controls > cases
AUC: 0.8976912

Fold: Fold09
Accuracy: 0.8249242
Sensitivity: 0.5580808
Specificity: 0.9548995
Setting direction: controls > cases
AUC: 0.8914956

Fold: Fold10
Accuracy: 0.8257993
Sensitivity: 0.5820017
Specificity: 0.9446494
Setting direction: controls > cases
AUC: 0.8984752

MEAN AUC BASED KFOLD



XGBOOST BASED HOLDOUT

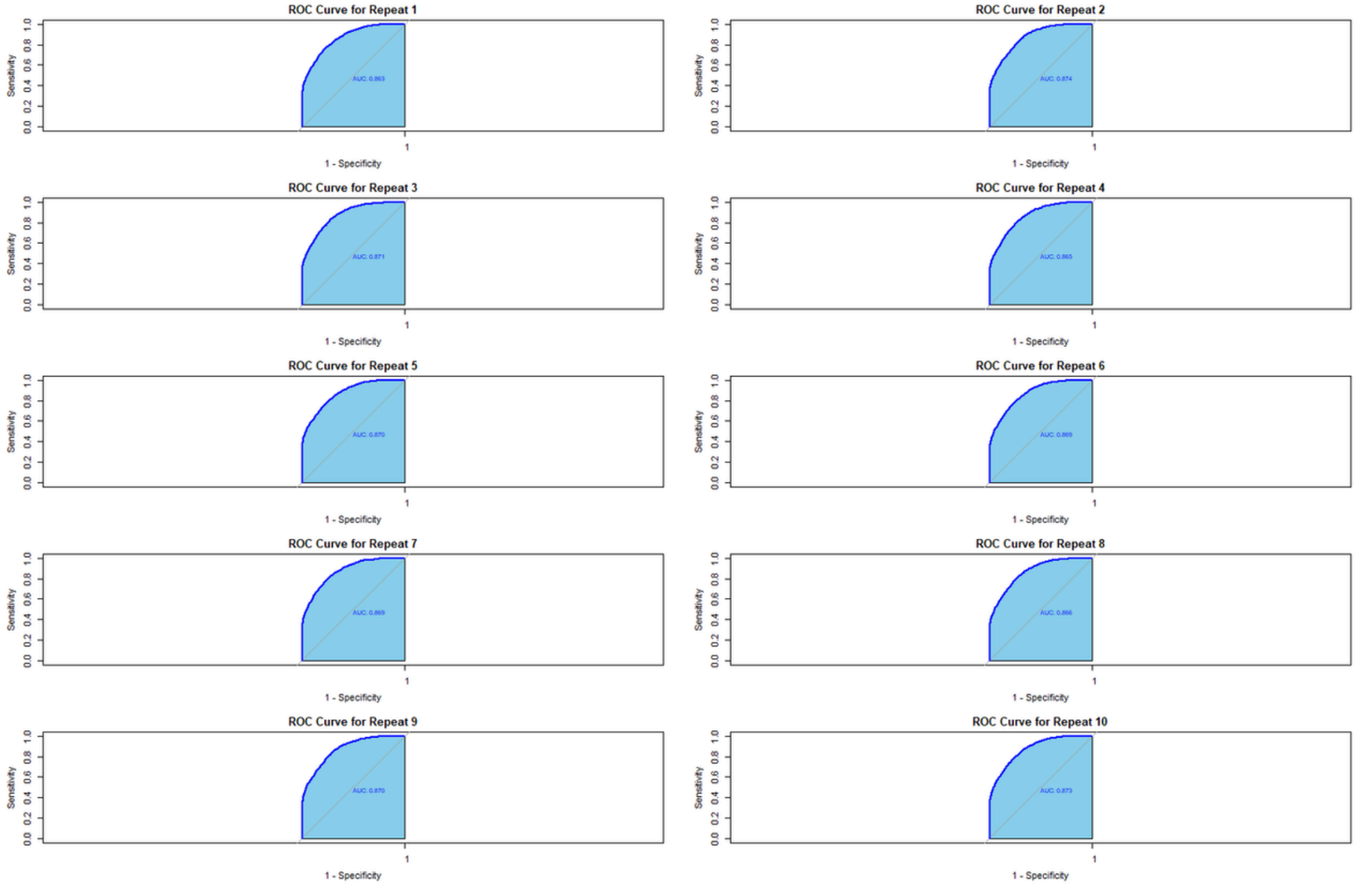
Repeat: 1
Accuracy: 0.8005146
Sensitivity: 0.4784303
Specificity: 0.9592593
Setting direction: controls > cases
AUC: 0.8629917

Repeat: 2
Accuracy: 0.8074979
Sensitivity: 0.5015497
Specificity: 0.9555495
Setting direction: controls > cases
AUC: 0.8741552

Repeat: 3
Accuracy: 0.8047413
Sensitivity: 0.4894406
Specificity: 0.9544654
Setting direction: controls > cases
AUC: 0.8713958

Repeat: 4
Accuracy: 0.7981255
Sensitivity: 0.4955971
Specificity: 0.9497862
Setting direction: controls > cases
AUC: 0.8651496

Repeat: 5
Accuracy: 0.8128273
Sensitivity: 0.5114959
Specificity: 0.9570652
Setting direction: controls > cases
AUC: 0.8697298



Repeat: 6
Accuracy: 0.805017
Sensitivity: 0.4949467
Specificity: 0.9558803
Setting direction: controls > cases
AUC: 0.8690133

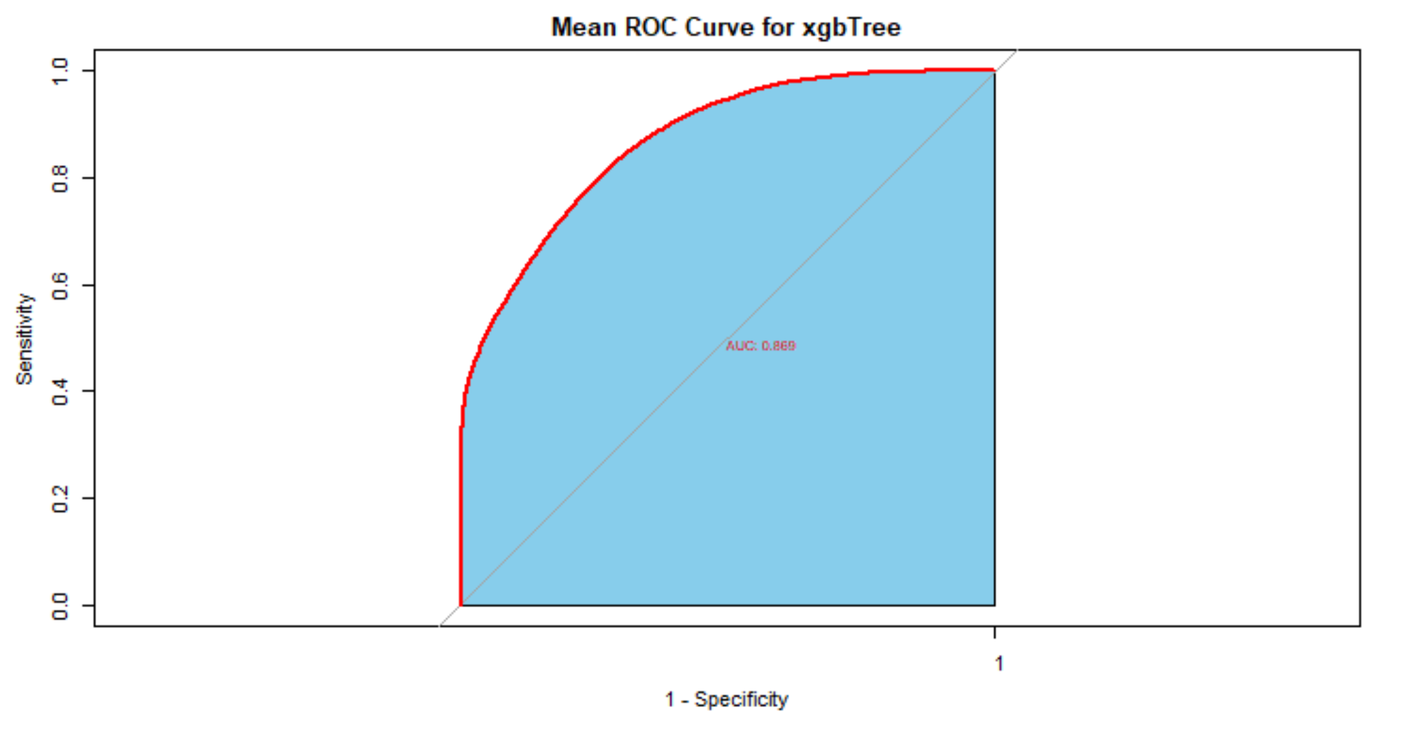
Repeat: 7
Accuracy: 0.80419
Sensitivity: 0.4911492
Specificity: 0.956308
Setting direction: controls > cases
AUC: 0.8693857

Repeat: 8
Accuracy: 0.8006983
Sensitivity: 0.4914971
Specificity: 0.9527138
Setting direction: controls > cases
AUC: 0.8660455

Repeat: 9
Accuracy: 0.8085087
Sensitivity: 0.5139431
Specificity: 0.9532685
Setting direction: controls > cases
AUC: 0.8704666

Repeat: 10
Accuracy: 0.8078655
Sensitivity: 0.504738
Specificity: 0.9569568
Setting direction: controls > cases
AUC: 0.872921

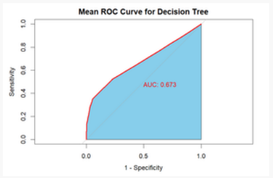
MEAN AUC BASED HOLDOUT

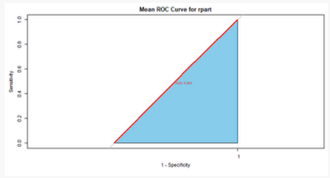



```
AUC values for each repeat:
 [1] 0.8629917 0.8741552 0.8713958 0.8651496 0.8697298 0.8690133
 [7] 0.8693857 0.8660455 0.8704666 0.8729210

Mean AUC: 0.8691254
```

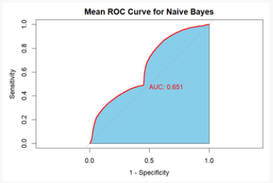
PERBANDINGAN MODEL

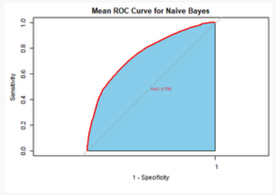
Decision Tree



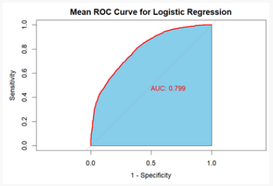


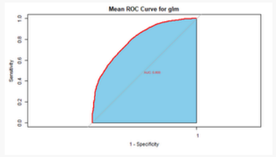
Naive Bayes



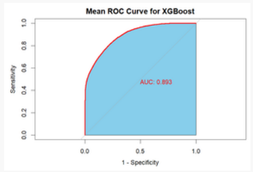


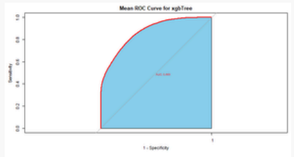
Regresi Logistik





XGBoost





- Model Regresi Logistik dan Model XGBoost menunjukkan kinerja yang konsisten baik dan cukup baik pada kedua teknik evaluasi (Repeated Holdout dan K-Fold). Ini menandakan bahwa kedua model ini stabil dan dapat diandalkan dalam berbagai skenario evaluasi.
- Model Naive Bayes menunjukkan kinerja yang cukup baik dengan Repeated Holdout tetapi hanya moderat dengan K-Fold.
- Model Decision Tree menunjukkan kinerja yang moderat dengan K-Fold tetapi buruk dengan Repeated Holdout.

Kesimpulan

Model XGBoost dan Model Regresi Logistik adalah model yang terpilih karena menunjukkan kinerja yang konsisten baik dan cukup baik pada kedua teknik evaluasi baik K-Fold maupun Holdout. Antara keduanya, XGBoost lebih unggul jika kita mempertimbangkan kinerja sangat baik pada K-Fold, yang sering dianggap lebih robust dalam mengatasi data imbalance.

Rekomendasi Akhir

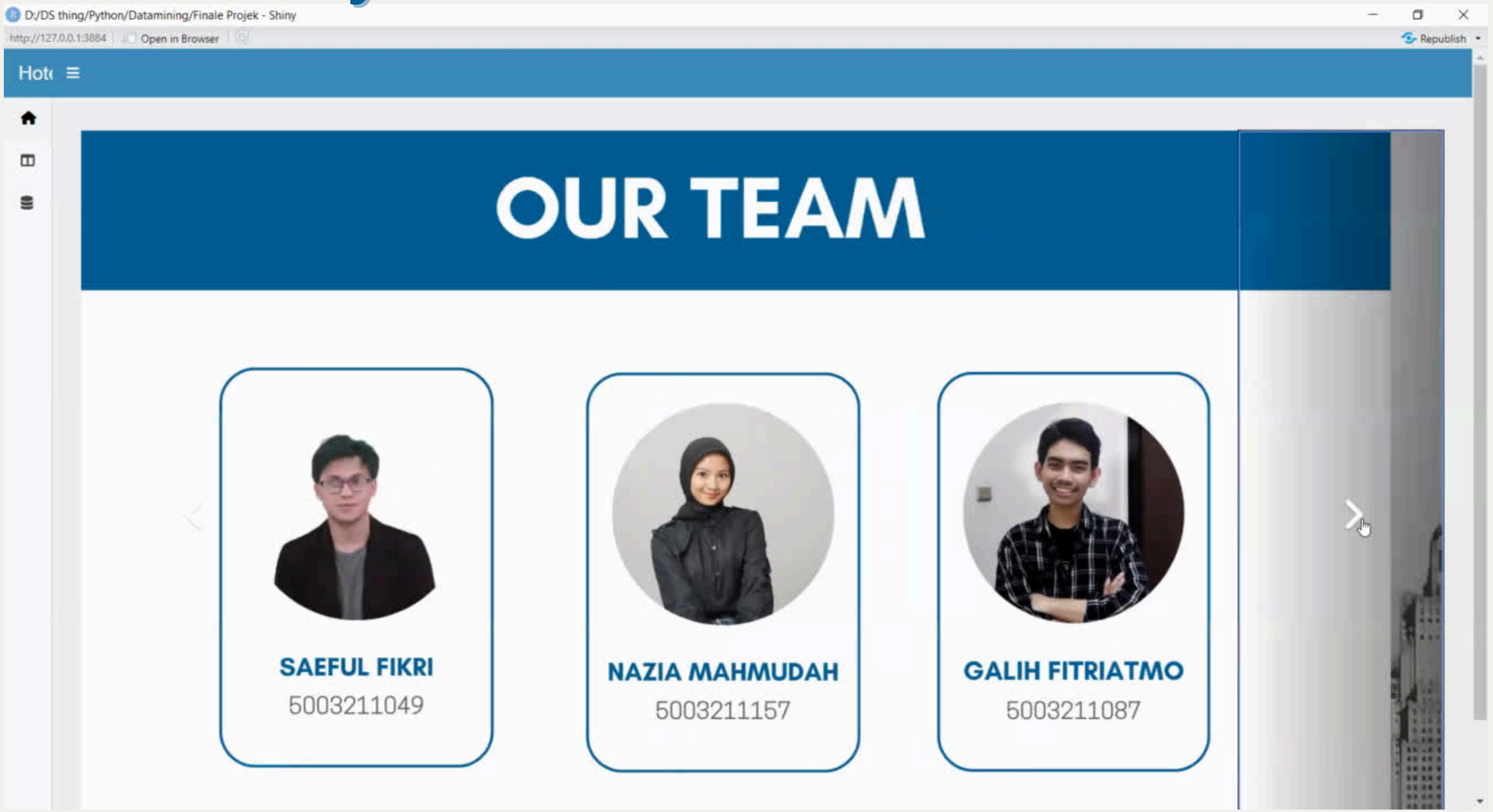
Model XGBoost adalah pilihan terbaik karena :

- Menunjukkan kinerja sangat baik dengan K-Fold dengan mean AUC 0.893.
- Menunjukkan kinerja baik dengan Repeated Holdout dengan mean AUC 0.869.
- Memiliki stabilitas dan keandalan yang baik dalam berbagai skenario evaluasi.
- Model Regresi Logistik juga merupakan pilihan yang sangat baik dan dapat dipertimbangkan sebagai alternatif yang solid jika kesederhanaan dan interpretabilitas lebih diutamakan.

Demo Time!

Demo R Shiny

https://bit.ly/DemoShinyKelompok1



thank you