IYKRA Training:

Model Evaluation Assignment

Nama: Anugrah Yudha Pranata

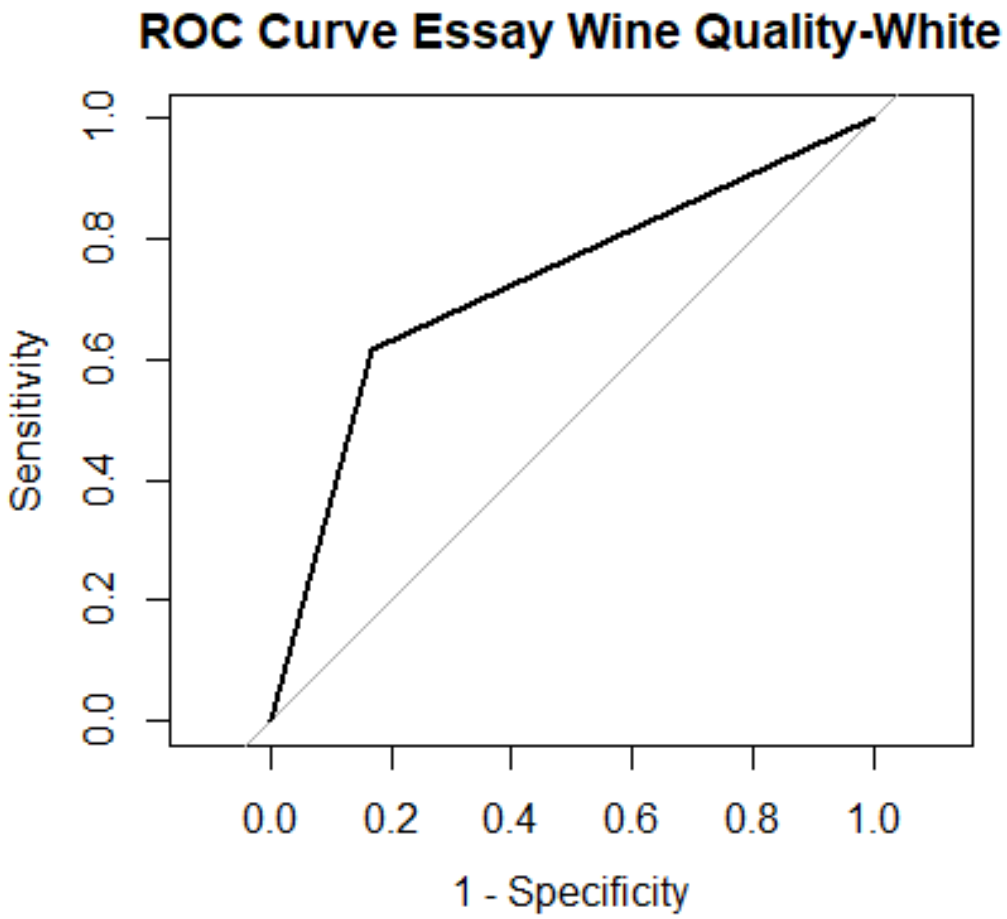
E-mail: anugrah.yudha150796@gmail.com

ANSWER TO QUESTION 1 - R CODE

```
# Loading Dataset 5: Essay Wine Quality White
df5 <- read.csv("essay_winequality-white.csv", sep = ";")
head(df5)
summary(df5)
# Untuk menjawab pertanyaan mengenai Lift dan Gain Chart, dataset 5 disimpan dengan nama temp (temporary)
temp <- df5

# Penentuan kelas berdasarkan variabel quality
df5$quality <- ifelse(df5$quality >= 6, "good", "bad")
df5$quality <- factor(df5$quality)

# Membagi dataset menjadi 80% training data dan 20% testing data
set.seed(12345)
df5.sample = sample.split(df5$quality,SplitRatio = 0.8)
df5.train = df5[df5.sample,]
df5.test = df5[!df5.sample,]

# Pembuatan model dengan menggunakan decision tree
df5.dtree = rpart(quality~.,data=df5.train)
# Tanpa memperhatikan parameter cp
df5.dtree
summary(df5.dtree)

# Melakukan prediksi dengan model yang sudah dibuat (Decision tree, variabel yang diprediksi bertipe data class)
df5.test.predict = predict(df5.dtree, newdata=df5.test, type="class")

# Confusion Matrix
caret::confusionMatrix(data=df5.test.predict, reference=df5.test$quality, positive = "good")
```

 # Didapatkan nilai accuracy adalah 0.7592

ANSWER TO QUESTION 1 - R CODE



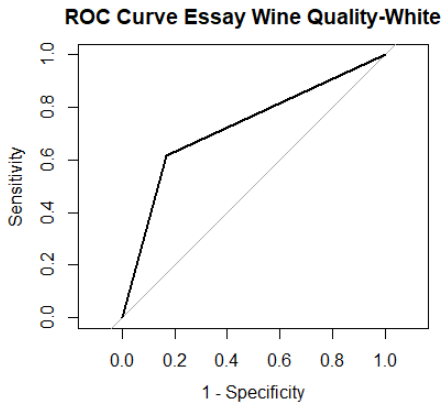
ROC Curve with default parameter (AUC value = 0.7236)

```
# Mengubah skala hasil prediksi dari class (kategorikal) menjadi numerik, untuk menghitung besarnya ROC dan AUC
df5.predictions <- ifelse(df5.test.predict=="good", 1, 0)
df5.labels <- ifelse(df5.test$quality=="good", 1, 0) # Labels sebagai reference

# Penghitungan ROC, AUC
pred.val3 <- prediction(df5.predictions,df5.labels)
auc.perf3 = performance(pred.val3, measure = "auc")
auc.perf3@y.values
perf2 = performance(pred.val3,measure="tpr",x.measure = "fpr")
#plot(perf2)
rocCurve.df5 <- roc(response = df5.labels, predictor = df5.predictions, levels = rev(levels(factor(df5.labels))))
auc(rocCurve.df5)
plot(rocCurve.df5,legacy.axes=TRUE)
title("ROC Curve Essay Wine Quality-White",line=+3)
```

 # Didapatkan hasil bahwa nilai AUC adalah 0.7236

Nilai AUC 0.7236 dapat dinyatakan sebagai cukup (fair)

ANSWER TO QUESTION 2 - R CODE



ROC Curve with default parameter
(AUC value = 0.7236)

```
# Melakukan Cross-Validation dengan K-fold sebanyak 10
ctrl = trainControl(method = "cv", number = 10)
grid = expand.grid(cp=seq(0,0.5,0.001))
# Diatur sequential dari nilai 0 hingga 0.5 dengan kenaikan incremental setiap 0.001
df5.tree.kcv.grid = train(quality~.,
                          data = df5.train, # Membuat model dengan menggunakan data training
                          method = "rpart", # Menggunakan method decision tree
                          trControl = ctrl,
                          tuneGrid = grid)
```

```
# Hasil grid
df5.tree.kcv.grid
```

```
Accuracy was used to select the optimal model using the largest value.
The final value used for the model was cp = 0.002.
```

cp value	accuracy
0.002	0.7588186
0.008638211	0.7462916

 # Berdasarkan hasil perhitungan grid, didapatkan bahwa nilai cp yang optimum adalah 0.002

Sebagai pembanding, dihitung juga: jika tidak menggunakan grid, berapakah cp optimum?

Hasilnya berbeda, tanpa grid didapatkan bahwa nilai cp yang optimum adalah 0.008xxx

 # Kemudian, berdasarkan nilai accuracy yang lebih besar, dipilih cp yang optimum adalah 0.002

ANSWER TO QUESTION 3 - R CODE

```
# Penyusunan model baru dengan tambahan parameter cp = 0.002
df5.dtree.cp = rpart(quality~.,data=df5.train, cp=0.002)
df5.dtree.cp

# Melakukan prediksi dengan model decision tree yang baru (yang ditambahkan dengan parameter cp)
df5.test.predict.cp = predict(df5.dtree.cp, newdata=df5.test, type="class")

# Confusion Matrix
caret::confusionMatrix(data=df5.test.predict.cp, reference=df5.test$quality, positive = "good")
```

 # Nilai Accuracy menjadi 0.7704

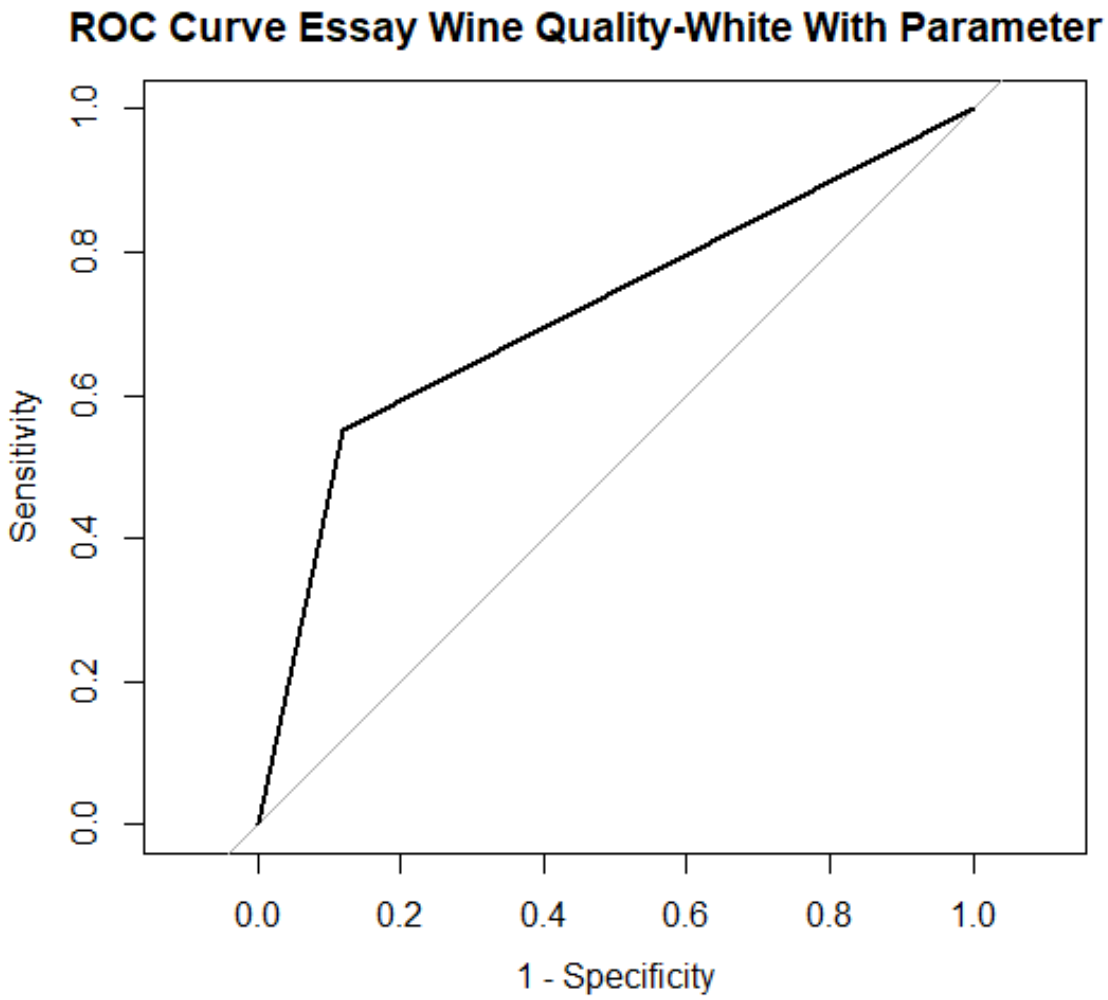
 # Kesimpulan: Nilai accuracy meningkat dengan cp = 0.002 dibandingkan tanpa mempertimbangkan parameter cp

```
# Untuk menghitung nilai ROC dan AUC, variabel prediksinya yang semula bertipe class (kategorikal), diubah menjadi skala numerik
df5.predictions.cp <- ifelse(df5.test.predict.cp=="good", 1, 0)

# Penghitungan ROC, AUC
pred.val3.cp <- prediction(df5.predictions.cp,df5.labels)
auc.perf3.cp = performance(pred.val3.cp, measure = "auc")
auc.perf3.cp@y.values
perf2.cp = performance(pred.val3.cp,measure="tpr",x.measure = "fpr")
#plot(perf2.cp)
rocCurve.df5.cp <- roc(response = df5.labels, predictor = df5.predictions.cp, levels = rev(levels(factor(df5.labels))))
auc(rocCurve.df5.cp)
plot(rocCurve.df5.cp,legacy.axes=TRUE)
title("ROC Curve Essay Wine Quality-White With Parameter cp",line=+3)
```

 # Namun, meskipun nilai akurasi meningkat, nilai AUC turun menjadi 0.7161 (berkurang 0.0075)

Kategori model tetap dalam kelompok “Cukup” (fair)

ANSWER TO QUESTION 3 - R CODE



ROC Curve with parameter cp = 0.002
(AUC value = 0.7161)

Parameter	Accuracy	AUC value	Kategori Model
Default Parameter	0.7592	0.7236	Cukup (fair)
CP = 0.002	0.7704	0.7161	Cukup (fair)

ANSWER TO QUESTION 3 - R CODE

 # Untuk membuat Lift Chart, Gain Chart, dan K-S, digunakan data temp yang sebelumnya telah kita simpan

```
# Pembuatan model dengan menggunakan decision tree, disesuaikan dengan cp yang telah didapatkan sebelumnya
temp.dtree.cp = rpart(quality~.,data=temp.train, cp = 0.002)
temp.dtree.cp
summary(temp.dtree.cp)
# Melakukan prediksi dengan model yang sudah dibuat (Decision tree, variabel yang diprediksi merupakan probabilitas)
temp.test.predict.cp = predict(temp.dtree.cp, newdata=temp.test)
# Mengkategorikan hasil sesuai treshold
temp.test.predict <- ifelse(temp.test.predict.cp >= 6, "good", "bad")
temp.test.reference <- ifelse(temp.test$quality >= 6, "good", "bad")

# Confusion Matrix
caret::confusionMatrix(data=temp.test.predict,
reference=temp.test.reference, positive = "good")
```

 # Didapatkan hasil bahwa nilai Accuracy menurun menjadi 0.6388

```
# Mengubah skala kategorik menjadi numerik untuk menghitung ROC, AUC
temp.test.predict.num <- ifelse(temp.test.predict=="good", 1, 0)
temp.test.reference.num <- ifelse(temp.test.reference=="good", 1, 0)

# Penghitungan ROC, AUC
pred.val4.cp <- prediction(temp.test.predict.num,temp.test.reference.num)
auc.perf4.cp = performance(pred.val4.cp, measure = "auc")
auc.perf4.cp@y.values
perf3.cp = performance(pred.val4.cp,measure="tpr",x.measure = "fpr")
#plot(perf3.cp)
rocCurve.temp.cp <- roc(response = temp.test.reference.num, predictor =
temp.test.predict.num, levels = rev(levels(factor(temp.test.reference.num))))
auc(rocCurve.temp.cp)
plot(rocCurve.temp.cp,legacy.axes=TRUE)
title("ROC Curve Essay Wine Quality-White With Parameter cp",line=+3)
```

 # Didapatkan hasil bahwa nilai AUC adalah 0.7058. Nilai AUC kembali turun daripada sebelumnya

```
# Kategori model tetap dalam kelompok “Cukup” (fair)
```

ANSWER TO QUESTION 3 - R CODE

```
## Membuat Lift Chart dan Gain Chart secara manual
temp.predict.actual = data.frame(temp.test$quality,temp.test.predict.cp) %>% arrange(-temp.test.predict.cp)
# arrange adalah untuk mengurutkan. Penamaan variabel sudah menyesuaikan dengan penggunaan baku dplyr/tidyverse
temp.decile = temp.predict.actual %>% mutate(decile.10 = ntile(temp.test.predict.cp,10)) %>% group_by(decile.10)
# Dibagi berdasarkan decile (dibagi 10)
temp.decile.summ = temp.decile %>% mutate(temp.test.quality=ifelse(temp.test.quality>=6,"good","bad")) %>%
  group_by(decile.10,temp.test.quality) %>% summarise(Pop=n()) %>% spread(temp.test.quality,Pop)
# mutate kalau good jadi True atau False-nya dihilangkan, karena datanya sudah merupakan numerik (bukan kategorikal)
# Summarise berdasarkan nilai 1 (True) atau 0 (False). Ditambahkan mutate agar nama column-nya menjadi character. Spread
dibuat sebagai deep layer untuk mengubah long table menjadi wide table
temp.decile.summ[is.na(temp.decile.summ)]=0 # Jika terdapat NA, diisi dengan nilai 0
temp.decile.summ = temp.decile.summ %>% mutate(Population=bad+good) # Ditambahkan column baru bernama Populasi

# Di-export ke CSV untuk dibuka dalam Ms. Excel
write.csv(temp.decile.summ,"D:/BODT Camp IYKRA/Materi/#21/assignment_df5_temp.csv",row.names = FALSE)
```

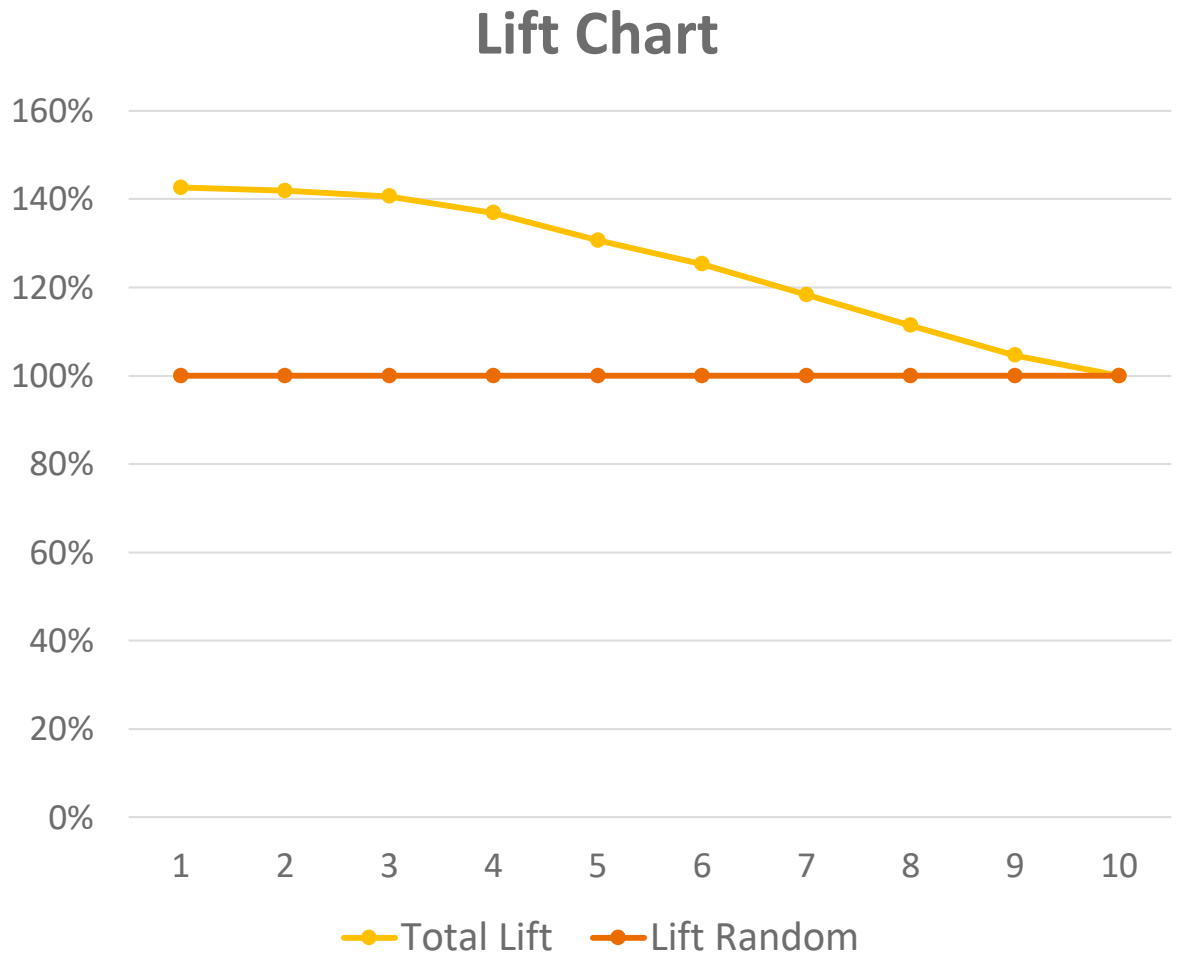
ANSWER TO QUESTION 3 - R CODE

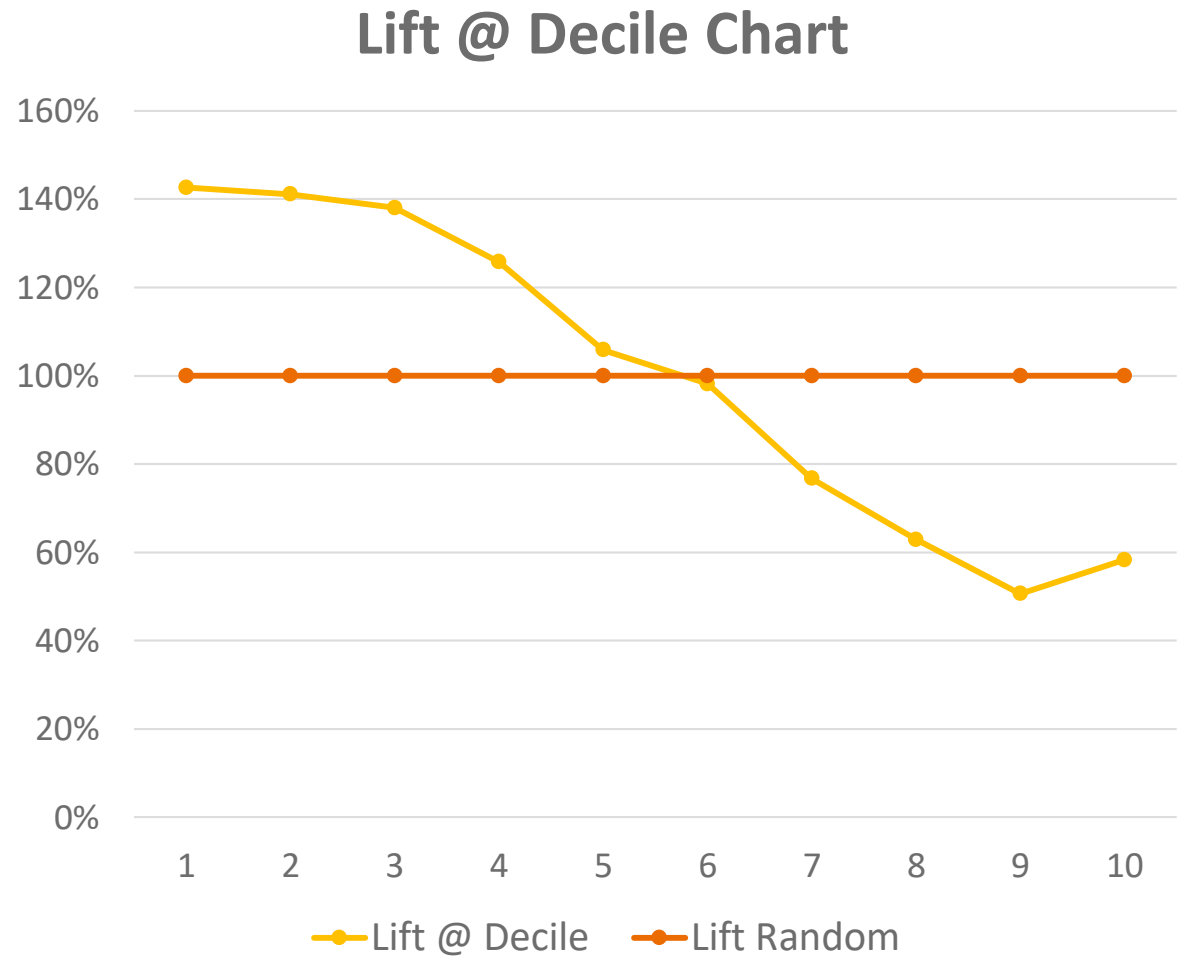
decile.10	good	bad	Population	%good	%bad	%pop	Cumm. good	Cumm. bad	Cumm. Pop	Total Lift	Lift @ Decile	Lift Random	K-S
1	93	5	98	1,52%	14,26%	10%	14,26%	1,52%	10%	143%	**143%**	100%	12,74%
2	92	6	98	1,83%	14,11%	10%	28,37%	3,35%	20%	142%	**141%**	100%	25,02%
3	90	8	98	2,44%	13,80%	10%	42,18%	5,79%	30%	141%	**138%**	100%	36,39%
4	82	16	98	4,88%	12,58%	10%	54,75%	10,67%	40%	137%	**126%**	100%	44,08%
5	69	29	98	8,84%	10,58%	10%	65,34%	19,51%	50%	131%	**106%**	100%	**45,83%**
6	64	34	98	10,37%	9,82%	10%	75,15%	29,88%	60%	125%	98%	100%	45,28%
7	50	48	98	14,63%	7,67%	10%	82,82%	44,51%	70%	118%	77%	100%	38,31%
8	41	57	98	17,38%	6,29%	10%	89,11%	61,89%	80%	111%	63%	100%	27,22%
9	33	65	98	19,82%	5,06%	10%	94,17%	81,71%	90%	105%	51%	100%	12,46%
10	38	60	98	18,29%	5,83%	10%	100,00%	100,00%	100%	100%	58%	100%	0,00%
TOTAL	652	328	980	100%	100%	100%							

Berdasarkan nilai Lift @ Decile, diketahui pula bahwa kita dapat percaya kepada model (model lebih baik daripada hasil random) hingga pada decile ke-5. Dikarenakan nilai Lift @ Decile yang lebih besar daripada nilai Lift Random (100%).

Lalu, didapatkan pula hasil dari Kolmogorov-Smirnov chart bahwa nilai K-S adalah 45,83% atau 0.4583.

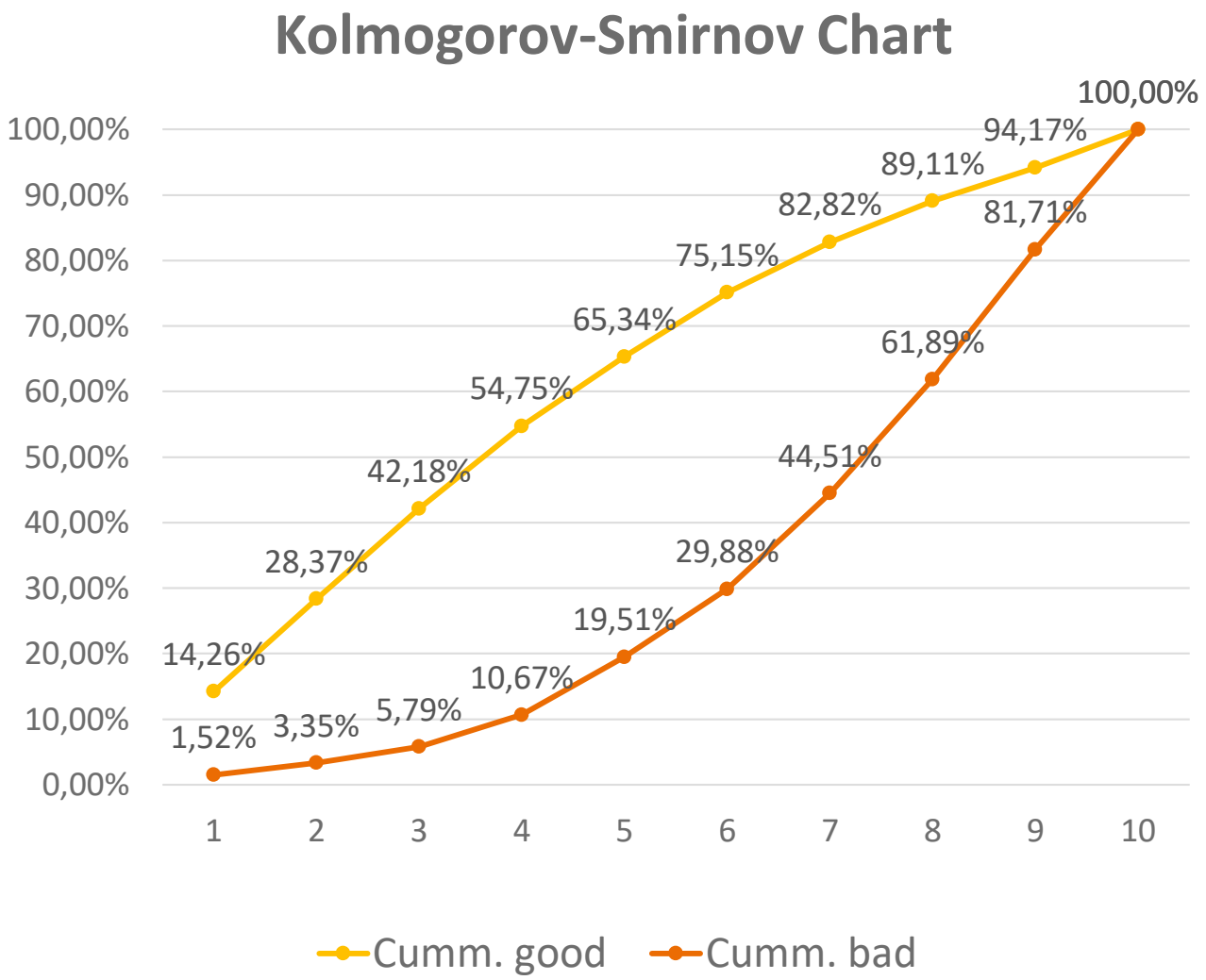
ANSWER TO QUESTION 3 - R CODE





Berdasarkan lift chart di atas, dapat dinyatakan bahwa kita dapat percaya kepada model (model lebih baik daripada hasil random) hingga pada decile ke-5. Dikarenakan nilai Lift @ Decile yang lebih besar daripada nilai Lift Random (100%).

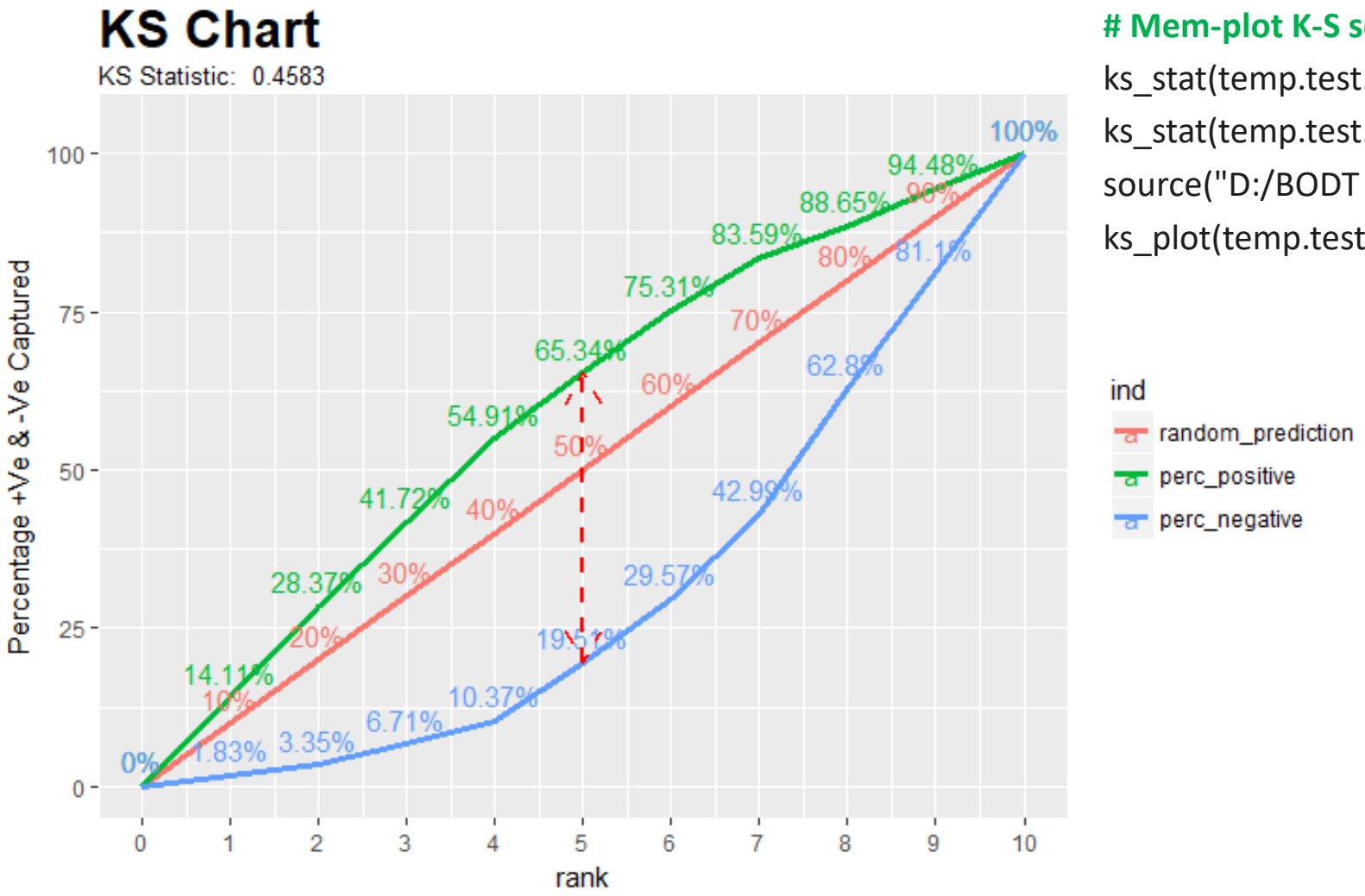
ANSWER TO QUESTION 3 - R CODE



Hasil dari Kolmogorov-Smirnov chart berikut ini adalah bahwa nilai K-S adalah 45,83% atau 0.4583, tepatnya pada decile ke-5.

Nilai K-S = 0.4583 berarti model dinilai kurang mampu untuk membedakan hasil (prediksi) yang baik dan buruk (lebih kecil dari 0.5)

ANSWER TO QUESTION 3 - R CODE




```
# Mem-plot K-S secara otomatis
ks_stat(temp.test.reference,temp.test.predict.cp, returnKSTable = T)
ks_stat(temp.test.reference,temp.test.predict.cp)
source("D:/BODT Camp IYKRA/Materi/#21/KS_Plot_Function.R")
ks_plot(temp.test.reference,temp.test.predict.cp)
```