UAS PSB

Nana Oktaviana

2023-06-13

Diberikan data tentang informasi yang diperoleh dari 150 artikel-artikel berita online. Di dalam data tersebut terdapat label yang menunjukkan apakah artikel-artikel berita merupakan berita yang asli atau palsu. Data terdiri dari 4 kolom yaitu : (1) kata_dalam_artikel, (2) tanda_seru_judul_artikel, (3) persen_negatif, dan (4) tipe_berita. Berikut adalah deskripsi dari masing masing kolom

Nama Kolom	Deskripsi
kata_dalam_artikel	banyaknya kata dalam artikel
tanda_seru_judul_artikel	apakah judul artikel mengandung tanda seru?
persen_negatif	persentase kata-kata yang memiliki sentimen negatif
tipe_berita	label artikel berita untuk berita asli atau palsu

```
library(readxl)
dt.ss <- read_excel("C:/Users/ASUS/Downloads/berita_palsu.xlsx",sheet=1)
str(dt.ss)
```

```
## tibble [150 × 4] (S3: tbl_df/tbl/data.frame)
##  $ kata_dalam_artikel      : num [1:150] 219 509 494 268 479 220 184 500
677 485 ...
##  $ tanda_seru_judul_artikel: chr [1:150] "tidak" "tidak" "ya" "tidak" ...
##  $ persen_negatif          : num [1:150] 8.47 4.74 3.33 6.09 2.66 3.02 4.1
4.63 2.18 4.22 ...
##  $ tipe_berita             : chr [1:150] "palsu" "asli" "palsu" "asli" ...
```

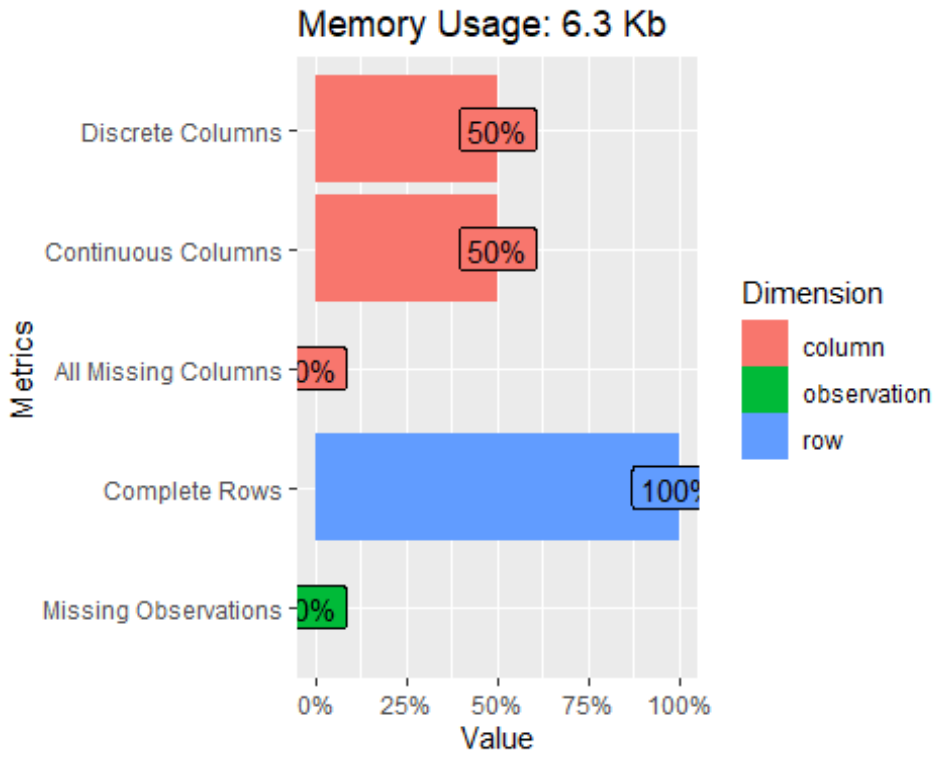
```
View(dt.ss)
```

```
##kategori "tidak" -> "1", "ya" -> "2"
dt.ss$tanda_seru_judul_artikel[dt.ss$tanda_seru_judul_artikel=="tidak"] <- 1
dt.ss$tanda_seru_judul_artikel[dt.ss$tanda_seru_judul_artikel=="ya"] <- 2
dt.ss$tanda_seru_judul_artikel <- as.factor(dt.ss$tanda_seru_judul_artikel)
##kategori "palsu" -> "1", "asli" -> "2"
dt.ss$tipe_berita[dt.ss$tipe_berita=="palsu"] <- 1
dt.ss$tipe_berita[dt.ss$tipe_berita=="asli"] <- 2
dt.ss$tipe_berita <- as.factor(dt.ss$tipe_berita)
str(dt.ss)
```

```
## tibble [150 × 4] (S3: tbl_df/tbl/data.frame)
##  $ kata_dalam_artikel      : num [1:150] 219 509 494 268 479 220 184 500
677 485 ...
##  $ tanda_seru_judul_artikel: Factor w/ 2 levels "1","2": 1 1 2 1 1 1 1 2 1
1 ...
##  $ persen_negatif          : num [1:150] 8.47 4.74 3.33 6.09 2.66 3.02 4.1
4.63 2.18 4.22 ...
##  $ tipe_berita             : Factor w/ 2 levels "1","2": 1 2 1 2 1 2 1 1 1
2 ...

View(dt.ss)
```

A. Eksplorasi Data

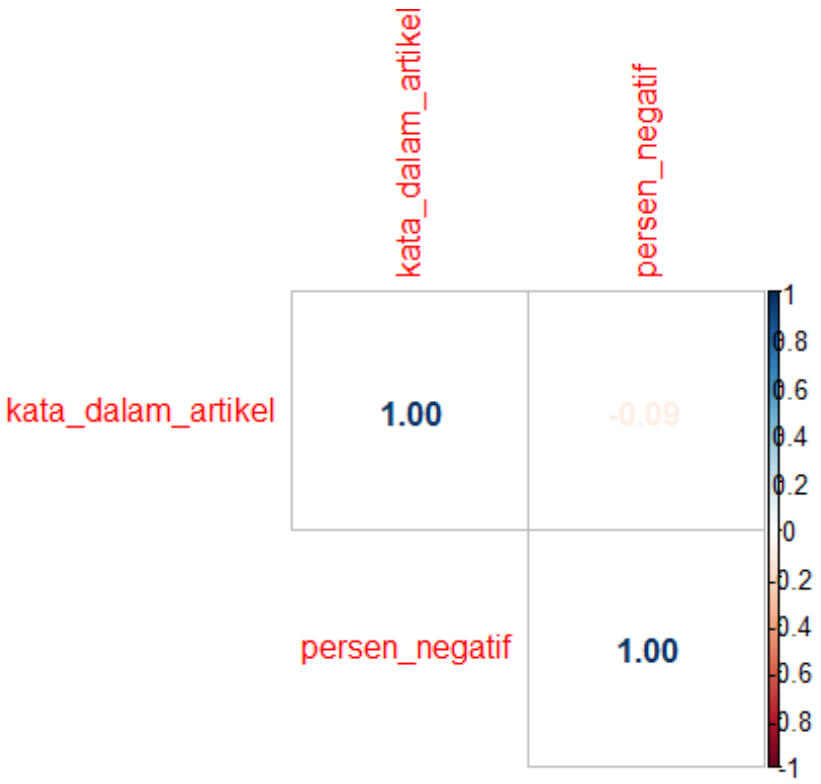
```
library(DataExplorer)
plot_intro(data = dt.ss)
```



Pengecekan proporsi tipe data dan *missing value* menunjukkan bahwa data yang digunakan merupakan data lengkap karena jumlah baris dan kolom berjumlah 100%. Tipe data pada kasus ini yaitu diskrit dan kontinu. Selain itu, data yang digunakan tidak terdeteksi *missing value* atau data hilang.

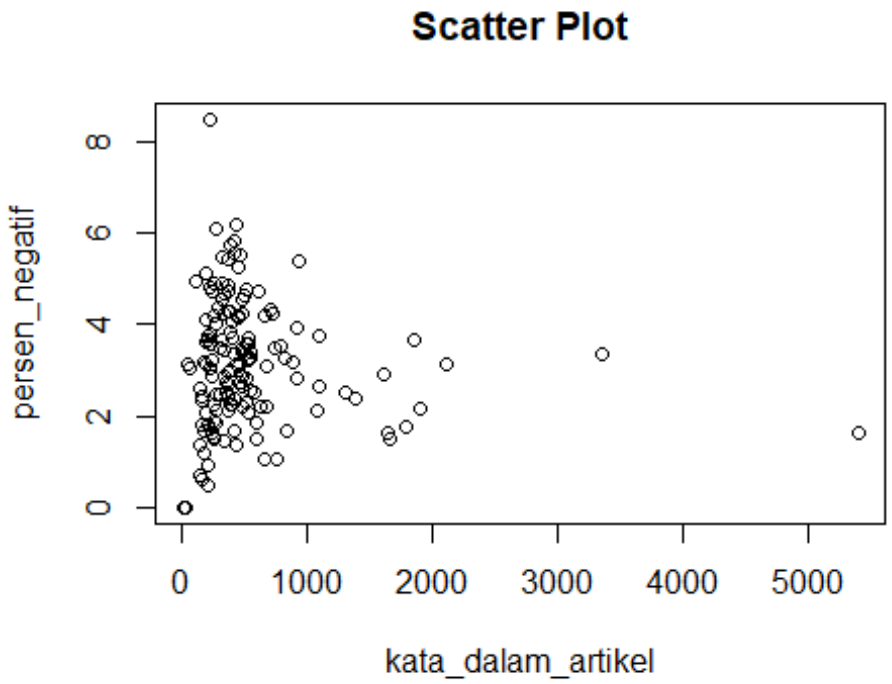
```
library(corrplot)

## corrplot 0.92 loaded
```

```
korel<-cor(dt.ss[,-c(2,4)])
corrplot(korel, type ="upper", method="number")
```

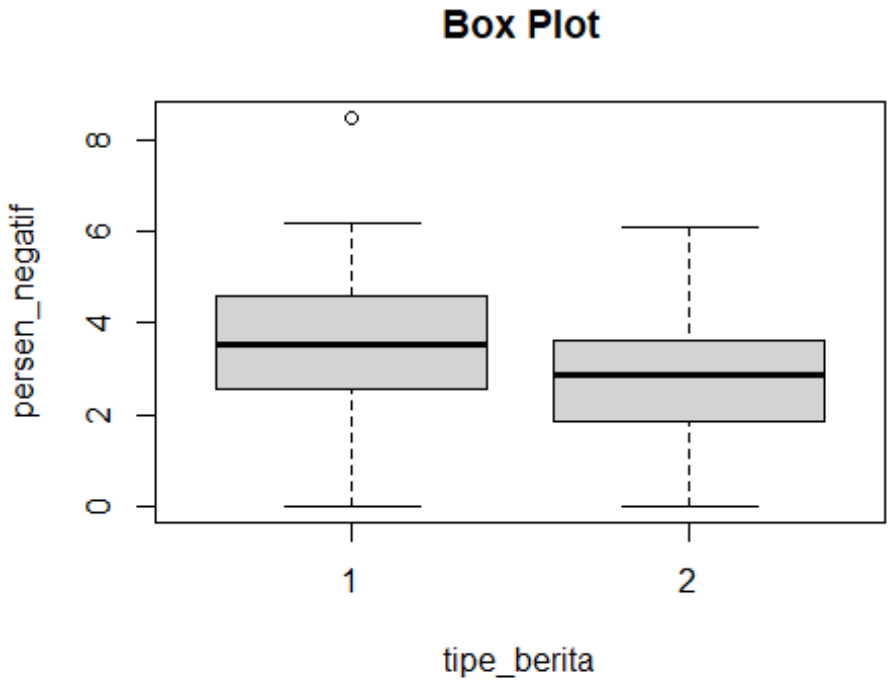


Berdasarkan output tersebut, diketahui bahwa kata_dalam_artikel memiliki korelasi negatif dengan nilai yang rendah.

```
# Scatter plot: kata_dalam_artikel vs. persen_negatif
plot(dt.ss$kata_dalam_artikel, dt.ss$persen_negatif, xlab =
"kata_dalam_artikel", ylab = "persen_negatif", main = "Scatter Plot")
```




```r
# Box plot: tipe_berita vs. persen_negatif
boxplot(dt.ss$persen_negatif ~ dt.ss$tipe_berita, xlab = "tipe_berita", ylab
= "persen_negatif", main = "Box Plot")
```



```
# Uji Chi-Square
chisq.test(dt.ss$tipe_berita, dt.ss$tanda_seru_judul_artikel)

##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  dt.ss$tipe_berita and dt.ss$tanda_seru_judul_artikel
## X-squared = 18.121, df = 1, p-value = 2.073e-05
```

Berdasarkan hasil uji Chi-Square ini, kita dapat menyimpulkan bahwa terdapat hubungan yang signifikan antara variabel kategorikal `tipe_berita` dan `tanda_seru_judul_artikel`

#B. Pembagian data

```
library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

set.seed(123)

# Bagi data menjadi data train (80%) dan data test (20%)
train_indices <- createDataPartition(dt.ss$tipe_berita, p = 0.8, list =
FALSE)
train_data <- dt.ss[train_indices, ]
test_data <- dt.ss[-train_indices, ]
```

#C. Model Naive Bayes

```
library(e1071)

# Model nb1: tipe_berita ~ tanda_seru_judul_artikel
model_nb1 <- naiveBayes(tipe_berita ~ tanda_seru_judul_artikel, data =
train_data)
summary(model_nb1)

##           Length Class  Mode
## apriori   2      table  numeric
## tables    1      -none- list
## levels    2      -none- character
## isnumeric 1      -none- logical
## call      4      -none- call

model_nb1

##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
```

```
## A-priori probabilities:
## Y
##   1   2
## 0.4 0.6
##
## Conditional probabilities:
##    tanda_seru_judul_artikel
## Y            1          2
##   1 0.75000000 0.25000000
##   2 0.97222222 0.02777778

summary(model_nb1)

##           Length Class  Mode
## apriori   2      table  numeric
## tables    1      -none- list
## levels    2      -none- character
## isnumeric 1      -none- logical
## call      4      -none- call

# Model nb2: tipe_berita ~ kata_dalam_artikel
model_nb2 <- naiveBayes(tipe_berita ~ kata_dalam_artikel, data = train_data)
model_nb2

##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##   1   2
## 0.4 0.6
##
## Conditional probabilities:
##    kata_dalam_artikel
## Y       [,1]     [,2]
##   1 455.1250 296.8904
##   2 604.8889 786.7288

summary(model_nb2)

##           Length Class  Mode
## apriori   2      table  numeric
## tables    1      -none- list
## levels    2      -none- character
## isnumeric 1      -none- logical
## call      4      -none- call
```

```r
# Model nb3: tipe_berita ~ kata_dalam_artikel + persen_negatif
model_nb3 <- naiveBayes(tipe_berita ~ kata_dalam_artikel + persen_negatif,
data = train_data)
model_nb3
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##   1   2
## 0.4 0.6
##
## Conditional probabilities:
##    kata_dalam_artikel
## Y       [,1]     [,2]
##   1 455.1250 296.8904
##   2 604.8889 786.7288
##
##    persen_negatif
## Y       [,1]     [,2]
##   1 3.515833 1.473854
##   2 2.670000 1.120620
```

```r
summary(model_nb3)
```

```
##           Length Class  Mode
## apriori   2      table  numeric
## tables    2      -none- list
## levels    2      -none- character
## isnumeric 2      -none- logical
## call      4      -none- call
```

```r
# Model nb4: tipe_berita ~ kata_dalam_artikel + persen_negatif +
tanda_seru_judul_artikel
model_nb4 <- naiveBayes(tipe_berita ~ kata_dalam_artikel + persen_negatif +
tanda_seru_judul_artikel, data = train_data)
model_nb4
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##   1   2
```

```
## 0.4 0.6
##
## Conditional probabilities:
##    kata_dalam_artikel
## Y       [,1]     [,2]
##   1 455.1250 296.8904
##   2 604.8889 786.7288
##
##    persen_negatif
## Y       [,1]     [,2]
##   1 3.515833 1.473854
##   2 2.670000 1.120620
##
##    tanda_seru_judul_artikel
## Y            1          2
##   1 0.75000000 0.25000000
##   2 0.97222222 0.02777778
```

```
summary(model_nb4)
```

```
##           Length Class  Mode
## apriori   2      table  numeric
## tables    3      -none- list
## levels    2      -none- character
## isnumeric 3      -none- logical
## call      4      -none- call
```

D. Prior

1. Hitung jumlah kemunculan kelas “1” dalam data train

2. Hitung jumlah kemunculan kelas “2” dalam data train

3. Hitung total jumlah observasi dalam data train

4. Hitung peluang prior untuk masing-masing kelas

prior_class1 <- count_class1 / total_count = 48/120 = 0.4

prior_class2 <- count_class2 / total_count = 72/120 = 0.6

E. Asumsi sebaran P(x|y)

Berdasarkan output yang Anda berikan, asumsi yang digunakan untuk sebaran P(x|y) pada model_nb3 adalah sebagai berikut:

Untuk peubah prediktor “kata_dalam_artikel”:

- P(x=kata_dalam_artikel|y=1) adalah distribusi Normal dengan rata-rata 455.1250 dan deviasi standar 296.8904.
- P(x=kata_dalam_artikel|y=2) adalah distribusi Normal dengan rata-rata 604.8889 dan deviasi standar 786.7288.

Untuk peubah prediktor “persen_negatif”:

- P(x=persen_negatif|y=1) adalah distribusi Normal dengan rata-rata 3.515833 dan deviasi standar 1.473854.
- P(x=persen_negatif|y=2) adalah distribusi Normal dengan rata-rata 2.670000 dan deviasi standar 1.120620.

Dalam Naive Bayes Classifier, asumsi yang umum digunakan adalah asumsi bahwa peubah prediktor terdistribusi secara independen, tetapi dalam kasus ini, terlihat bahwa distribusi yang digunakan adalah distribusi Normal (Gaussian) untuk masing-masing peubah prediktor.

Alasan di balik penggunaan asumsi distribusi Normal adalah karena Naive Bayes Classifier untuk prediktor diskrit ini menggunakan model distribusi Normal untuk memodelkan hubungan antara peubah prediktor dan peubah respon. Meskipun asumsi distribusi Normal ini sering digunakan, penting untuk memastikan bahwa asumsi ini sesuai dengan karakteristik data yang digunakan dan konteks masalah yang spesifik.

F. Syntax

```
# Membuat model_nb3 dengan asumsi distribusi Normal
model_nb3 <- naiveBayes(tipe_berita ~ kata_dalam_artikel + persen_negatif,
data = train_data, distribution = "gaussian")
model_nb3

##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace, distribution =
"gaussian")
##
## A-priori probabilities:
## Y
##   1   2
## 0.4 0.6
##
## Conditional probabilities:
##    kata_dalam_artikel
## Y       [,1]     [,2]
##   1 455.1250 296.8904
##   2 604.8889 786.7288
##
##    persen_negatif
```

```
## Y       [,1]     [,2]
##   1 3.515833 1.473854
##   2 2.670000 1.120620
```

```
# Menampilkan ringkasan model
summary(model_nb3)
```

```
##           Length Class  Mode
## apriori   2      table  numeric
## tables    2      -none- list
## levels    2      -none- character
## isnumeric 2      -none- logical
## call      5      -none- call
```

G. Model Regresi Logistik

```
# Model rl1: tipe_berita ~ tanda_seru_judul_artikel
model_rl1 <- glm(tipe_berita ~ tanda_seru_judul_artikel, data = train_data,
family = "binomial")
summary(model_rl1)
```

```
##
## Call:
## glm(formula = tipe_berita ~ tanda_seru_judul_artikel, family = "binomial",
##     data = train_data)
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  0.6650     0.2051   3.242  0.00119 **
## tanda_seru_judul_artikel2   -2.4567     0.7908  -3.107  0.00189 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 161.52  on 119  degrees of freedom
## Residual deviance: 147.33  on 118  degrees of freedom
## AIC: 151.33
##
## Number of Fisher Scoring iterations: 4
```

```
# Model rl2: tipe_berita ~ kata_dalam_artikel
model_rl2 <- glm(tipe_berita ~ kata_dalam_artikel, data = train_data, family
= "binomial")
summary(model_rl2)
```

```
##
## Call:
## glm(formula = tipe_berita ~ kata_dalam_artikel, family = "binomial",
##     data = train_data)
##
```

```
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)        0.1516359  0.2745746   0.552    0.581
## kata_dalam_artikel 0.0004933  0.0004150   1.189    0.235
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 161.52  on 119  degrees of freedom
## Residual deviance: 159.65  on 118  degrees of freedom
## AIC: 163.65
##
## Number of Fisher Scoring iterations: 4
```

```
# Model rl3: tipe_berita ~ kata_dalam_artikel + persen_negatif
model_rl3 <- glm(tipe_berita ~ kata_dalam_artikel + persen_negatif, data =
train_data, family = "binomial")
summary(model_rl3)
```

```
##
## Call:
## glm(formula = tipe_berita ~ kata_dalam_artikel + persen_negatif,
##     family = "binomial", data = train_data)
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)         1.7732898  0.5963891   2.973  0.00295 **
## kata_dalam_artikel  0.0004792  0.0004494   1.066  0.28635
## persen_negatif     -0.5261423  0.1660615  -3.168  0.00153 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 161.52  on 119  degrees of freedom
## Residual deviance: 147.79  on 117  degrees of freedom
## AIC: 153.79
##
## Number of Fisher Scoring iterations: 4
```

```
# Model rl4: tipe_berita ~ kata_dalam_artikel + persen_negatif +
tanda_seru_judul_artikel
model_rl4 <- glm(tipe_berita ~ kata_dalam_artikel + persen_negatif +
tanda_seru_judul_artikel, data = train_data, family = "binomial")
summary(model_rl4)
```

```
##
## Call:
## glm(formula = tipe_berita ~ kata_dalam_artikel + persen_negatif +
##     tanda_seru_judul_artikel, family = "binomial", data = train_data)
##
## Coefficients:
```

```
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 1.8603788  0.6177687   3.011  0.00260 **
## kata_dalam_artikel          0.0004344  0.0004618   0.941  0.34684
## persen_negatif             -0.4713723  0.1716289  -2.746  0.00602 **
## tanda_seru_judul_artikel2 -2.2473430  0.8123347  -2.767  0.00567 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 161.52  on 119  degrees of freedom
## Residual deviance: 137.26  on 116  degrees of freedom
## AIC: 145.26
##
## Number of Fisher Scoring iterations: 4
```

H. Asumsi Sebaran Prior

Berdasarkan model regresi logistik (model_rl4) yang telah dibuat, asumsi sebaran prior yang umum digunakan untuk intercept dan koefisien adalah sebagai berikut:

Asumsi sebaran prior untuk Intercept:

Intercept pada model regresi logistik biasanya diasumsikan memiliki sebaran prior yang terdistribusi secara Normal (Gaussian) dengan mean 0 dan varian yang cukup besar.
Asumsi sebaran prior untuk Koefisien (beta) pada peubah prediktor:

Koefisien pada model regresi logistik juga diasumsikan memiliki sebaran prior yang terdistribusi secara Normal dengan mean 0 dan varian yang cukup besar. Asumsi ini dipilih karena sebaran prior yang simetris di sekitar 0 memberikan fleksibilitas yang cukup besar pada model, memungkinkan model untuk menyesuaikan diri terhadap data yang diberikan.

I. Syntax

```r
# Mengatur prior untuk intercept
prior_intercept <- 0  # Mean = 0
```

J. Sensitivitas dan Spesifisitas

```r
# Menggunakan model_nb3 dengan data test
probabilities_nb3 <- predict(model_nb3, newdata = test_data, type = "class")

# Prediksi menggunakan model_rl3 dengan data test
probabilities_rl3 <- predict(model_rl3, newdata = test_data, type =
"response")
predictions_rl3 <- ifelse(probabilities_rl3 >= 0.5, "1", "2")
```

```r
# Definisikan fungsi untuk menghitung sensitivitas dan spesifisitas
calculate_sensitivity_specificity <- function(actual, predicted) {
  tp <- sum(actual == "1" & predicted == "1")  # True Positive
  tn <- sum(actual == "2" & predicted == "2")  # True Negative
  fp <- sum(actual == "2" & predicted == "1")  # False Positive
  fn <- sum(actual == "1" & predicted == "2")  # False Negative

  sensitivity <- tp / (tp + fn)  # Sensitivitas (True Positive Rate)
  specificity <- tn / (tn + fp)  # Spesifisitas (True Negative Rate)

  return(list(sensitivity = sensitivity, specificity = specificity))
}

# Menghitung sensitivitas dan spesifisitas untuk model_nb3 dengan cut-off
0.5, 0.6, dan 0.7
cutoffs <- c(0.5, 0.6, 0.7)  # Nilai cut-off peluang yang ingin digunakan
results_nb3 <- lapply(cutoffs, function(cutoff) {
  predictions <- ifelse(as.numeric(probabilities_nb3) >= cutoff, "1", "2")
  calculate_sensitivity_specificity(as.character(test_data$tipe_berita),
predictions)
})
results_nb3
```

```
## [[1]]
## [[1]]$sensitivity
## [1] 1
##
## [[1]]$specificity
## [1] 0
##
##
## [[2]]
## [[2]]$sensitivity
## [1] 1
##
## [[2]]$specificity
## [1] 0
##
##
## [[3]]
## [[3]]$sensitivity
## [1] 1
##
## [[3]]$specificity
## [1] 0
```

```r
# Menghitung sensitivitas dan spesifisitas untuk model_rl3 dengan cut-off
0.5, 0.6, dan 0.7
results_rl3 <- lapply(cutoffs, function(cutoff) {
  predictions <- ifelse(probabilities_rl3 >= cutoff, "1", "2")
```

```
  calculate_sensitivity_specificity(test_data$tipe_berita, predictions)
})
results_rl3
```

```
## [[1]]
## [[1]]$sensitivity
## [1] 0.4166667
##
## [[1]]$specificity
## [1] 0.2777778
##
##
## [[2]]
## [[2]]$sensitivity
## [1] 0.4166667
##
## [[2]]$specificity
## [1] 0.6666667
##
##
## [[3]]
## [[3]]$sensitivity
## [1] 0
##
## [[3]]$specificity
## [1] 0.7777778
```

J. Kombinasi Model Terbaik

Berdasarkan hasil di atas, tidak ada kombinasi model dan cut-off yang secara konsisten memiliki sensitivitas dan spesifisitas yang tinggi. Namun, jika kita lebih mengutamakan spesifisitas untuk memastikan bahwa artikel berita yang terdeteksi palsu adalah benar-benar palsu, maka kombinasi model_rl3 dengan cut-off 0.7 dapat menjadi pilihan terbaik.