Analisis Sentiment Pengguna Shopee Pada Fitur COD 2025 Dengan Regresi Logistik

Rafly Priyantama Ramadhan Bagaskara

2025-11-30

1. Load Library

```
library(readr)
library(dplyr)
library(caret)
library(pROC)
library(FSelectorRcpp)
library(glmnet)
```

2. Load Data

```
data_model <- read_csv("C:/Users/priya/Data Sentiment Ulasan Shopee COD 2025.csv")
```

```
## Rows: 2826 Columns: 6154
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (1): sentiment
## dbl (6153): aamiin, aamiin aamiin, aamiin robbal, aangat, aangat ahopee, aba...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(data_model)
```

```
## # A tibble: 6 x 6,154
##   sentiment aamiin `aamiin aamiin` `aamiin robbal` aangat `aangat ahopee`  abal
##   <chr>      <dbl>           <dbl>           <dbl>  <dbl>           <dbl> <dbl>
## 1 negative       0               0               0      0               0     0
## 2 negative       0               0               0      0               0     0
## 3 negative       0               0               0      0               0     0
## 4 negative       0               0               0      0               0     0
## 5 positive       0               0               0      0               0     0
## 6 negative       0               0               0      0               0     0
## # i 6,147 more variables: `abal abal` <dbl>, `abal cod` <dbl>,
## #   `abal tipu` <dbl>, abang <dbl>, `abang sengaja` <dbl>, abis <dbl>,
```

```
## #   `abis antar` <dbl>, `abis besok` <dbl>, `abis brg` <dbl>, acara <dbl>,
## #   acc <dbl>, account <dbl>, aceh <dbl>, `aceh cod` <dbl>, ada <dbl>,
## #   adain <dbl>, `adain cod` <dbl>, `adain qris` <dbl>, adakan <dbl>,
## #   `adakan cod` <dbl>, adek <dbl>, adil <dbl>, admintnya <dbl>, adu <dbl>,
## #   aduh <dbl>, af <dbl>, aga <dbl>, agam <dbl>, agen <dbl>, ...
```

3. Fungsi Model Logistik Regresi

```
glmnet_sentiment <- function(data,
                             train_prop = 0.8,
                             alpha = 1,
                             lambda = 0.01,
                             seed = 123) {
  # 1. Siapkan data
  data_model <- data %>%
    filter(sentiment %in% c("positive", "negative")) %>%
    mutate(
      sentiment = factor(sentiment,
                         levels = c("negative", "positive"))
    )

  y <- data_model$sentiment
  X <- data_model %>% select(-sentiment)

  stopifnot(nrow(X) == length(y))

  # 2. Train-test split
  set.seed(seed)
  n <- nrow(data_model)
  n_train <- floor(train_prop * n)
  idx_train <- sample(1:n, size = n_train)

  X_train <- X[idx_train, ]
  X_test  <- X[-idx_train, ]

  y_train <- y[idx_train]
  y_test  <- y[-idx_train]

  # jaga level faktor
  y_train <- factor(y_train, levels = c("negative", "positive"))
  y_test  <- factor(y_test,  levels = c("negative", "positive"))

  # 3. Feature selection (ig di data train)
  train_fs <- cbind(X_train, sentiment = y_train)

  ig_scores <- information_gain(sentiment ~ ., data = train_fs)
  ig_scores <- ig_scores[order(-ig_scores$importance), , drop = FALSE]

  # semua importance tidak = 0, pakai yg > 0
  if (any(ig_scores$importance > 0)) {
    selected_features <- ig_scores %>%
      filter(importance > 0) %>%
```

```
      pull(attributes)
  } else {
    selected_features <- ig_scores$attributes
  }

  X_train_sel <- X_train[, selected_features, drop = FALSE]
  X_test_sel  <- X_test[,  selected_features, drop = FALSE]

  # 4. GLMNET
  X_train_mat <- as.matrix(X_train_sel)
  X_test_mat  <- as.matrix(X_test_sel)

  model_glmnet <- glmnet(
    x = X_train_mat,
    y = y_train,
    family = "binomial",
    alpha = alpha,
    lambda = lambda
  )

  # 5. Prediksi & evaluasi
  pred_prob <- predict(
    model_glmnet,
    newx = X_test_mat,
    type = "response"
  )[, 1]

  pred_class <- ifelse(pred_prob > 0.5, "positive", "negative")
  pred_class <- factor(pred_class, levels = c("negative", "positive"))

  cm <- confusionMatrix(pred_class, y_test, positive = "positive")

  roc_obj <- roc(y_test, pred_prob, levels = c("negative", "positive"))
  auc_val <- auc(roc_obj)

  cat("AUC (test set):", as.numeric(auc_val), "\n")
  print(cm$table)

  # 6. Return semua yang penting
  list(
    model = model_glmnet,
    selected_features = selected_features,
    confusion_matrix = cm,
    auc = auc_val,
    y_test = y_test,
    pred_prob = pred_prob,
    pred_class = pred_class
  )
}
```

4. Hasil Model

```
hasil_glmnet <- glmnet_sentiment(data_model)
```

```
## Setting direction: controls < cases
```

```
## AUC (test set): 0.8665128
##           Reference
## Prediction negative positive
##   negative      325       69
##   positive       31      141
```

```
# Hasil Area Under Curve (AUC)
hasil_glmnet$auc
```

```
## Area under the curve: 0.8665
```

```
# Hasil Confusion Matrix
hasil_glmnet$confusion_matrix
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction negative positive
##   negative      325       69
##   positive       31      141
##
##                Accuracy : 0.8233
##                  95% CI : (0.7894, 0.8539)
##     No Information Rate : 0.629
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6069
##
##  Mcnemar's Test P-Value : 0.0002156
##
##             Sensitivity : 0.6714
##             Specificity : 0.9129
##          Pos Pred Value : 0.8198
##          Neg Pred Value : 0.8249
##              Prevalence : 0.3710
##          Detection Rate : 0.2491
##    Detection Prevalence : 0.3039
##       Balanced Accuracy : 0.7922
##
##        'Positive' Class : positive
##
```