Week 05: ML Pipeline

_gu_npe.tnnx__

2024-09-07

Load Package

```
library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────────────────────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.4     ✓ readr     2.1.5
## ✓ forcats   1.0.0     ✓ stringr   1.5.1
## ✓ ggplot2   3.5.0     ✓ tibble    3.2.1
## ✓ lubridate 1.9.3     ✓ tidyr     1.3.1
## ✓ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────────────────────────────────────────────────────────────────
── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(mlbench)
library(caret)

## Loading required package: lattice
##
## Attaching package: 'caret'
```

```
##
## The following object is masked from 'package:purrr':
##
##     lift
```

Load Data set

```
df <- read.csv("C:/Users/ธนานพ/Downloads/archive/tumor.csv")
#df
```

Check data type in df

```
str(df)
```

```
## 'data.frame':    683 obs. of  10 variables:
##  $ Clump.Thickness            : int  5 5 3 6 4 8 1 2 2 4 ...
##  $ Uniformity.of.Cell.Size    : int  1 4 1 8 1 10 1 1 1 2 ...
##  $ Uniformity.of.Cell.Shape   : int  1 4 1 8 1 10 1 2 1 1 ...
##  $ Marginal.Adhesion          : int  1 5 1 1 3 8 1 1 1 1 ...
##  $ Single.Epithelial.Cell.Size: int  2 7 2 3 2 7 2 2 2 2 ...
##  $ Bare.Nuclei                : int  1 10 2 4 1 10 10 1 1 1 ...
##  $ Bland.Chromatin            : int  3 3 3 3 3 9 3 3 1 2 ...
##  $ Normal.Nucleoli            : int  1 2 1 7 1 7 1 1 1 1 ...
##  $ Mitoses                    : int  1 1 1 1 1 1 1 1 5 1 ...
##  $ Class                      : int  2 2 2 2 2 4 2 2 2 2 ...
```

Change df$Class to factor (Categorial)

```
df$Class <- as.factor(df$Class)
str(df)
```

```
## 'data.frame':    683 obs. of  10 variables:
##  $ Clump.Thickness            : int  5 5 3 6 4 8 1 2 2 4 ...
##  $ Uniformity.of.Cell.Size    : int  1 4 1 8 1 10 1 1 1 2 ...
##  $ Uniformity.of.Cell.Shape   : int  1 4 1 8 1 10 1 2 1 1 ...
##  $ Marginal.Adhesion          : int  1 5 1 1 3 8 1 1 1 1 ...
```

```
##  $ Single.Epithelial.Cell.Size: int  2 7 2 3 2 7 2 2 2 2 ...
##  $ Bare.Nuclei                : int  1 10 2 4 1 10 10 1 1 1 ...
##  $ Bland.Chromatin            : int  3 3 3 3 3 9 3 3 1 2 ...
##  $ Normal.Nucleoli            : int  1 2 1 7 1 7 1 1 1 1 ...
##  $ Mitoses                    : int  1 1 1 1 1 1 1 1 5 1 ...
##  $ Class                      : Factor w/ 2 levels "2","4": 1 1 1 1 1 2 1 1 1 1 ...
```

Cross Validation

```
## Cross validation
ctrl <- trainControl(
  method = "cv",         # Cross-validation
  number = 10            # 10-fold cross-validation
)
```

Train Model

```
set.seed(123)  # For reproducibility

model <- train(
  Class ~ .,             # Predict 'Class' based on all other columns
  data = df,             # Use the 'Glass' dataset
  method = "naive_bayes",   # Naive Bayes classifier
  trControl = ctrl          # Cross-validation control
)

print(model)

## Naive Bayes
##
## 683 samples
##   9 predictor
##   2 classes: '2', '4'
##
```

```
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 614, 614, 615, 615, 615, 615, ...
## Resampling results across tuning parameters:
##
##   usekernel  Accuracy   Kappa
##   FALSE      0.9619778  0.9176216
##    TRUE      0.9649829  0.9233315
##
## Tuning parameter 'laplace' was held constant at a value of 0
## Tuning
##  parameter 'adjust' was held constant at a value of 1
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were laplace = 0, usekernel = TRUE
##  and adjust = 1.
```

Confustion Matrix

```
## Confustion Matrix
p <- predict(model, newdata = df)
confusionMatrix(p, df$Class, positive = "2")

## Confusion Matrix and Statistics
##
##           Reference
## Prediction   2   4
##          2 434  13
##          4  10 226
##
##                Accuracy : 0.9663
##                  95% CI : (0.9499, 0.9785)
##     No Information Rate : 0.6501
```

```
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9258
##
##  Mcnemar's Test P-Value : 0.6767
##
##             Sensitivity : 0.9775
##             Specificity : 0.9456
##          Pos Pred Value : 0.9709
##          Neg Pred Value : 0.9576
##              Prevalence : 0.6501
##          Detection Rate : 0.6354
##    Detection Prevalence : 0.6545
##       Balanced Accuracy : 0.9615
##
##        'Positive' Class : 2
##
```

```
ggplot() +
  geom_col(data = data.frame(Class = levels(df$Class),
                             counting_p = table(p)),
           aes(x = Class, y = table(p)),
           fill = 'red', alpha = 0.5, width = 0.4) +
  geom_col(data = data.frame(Class = levels(df$Class),
                             Count = table(df$Class)),
           aes(x = Class, y = table(df$Class)),
           fill = 'lightblue', alpha = 0.4, width = 0.4) +
  labs(title = "Distribution of Predicted vs. Actual Classes",
       x = "Class", y = "Count") +
  theme_minimal() +
```

```
  theme(
    plot.title = element_text(hjust = 0.5, size = 14, face = "bold"),
    axis.title = element_text(size = 12)
  )
```

```
## Don't know how to automatically pick scale for object of type <table>.
## Defaulting to continuous.
```