LKP 8
DATA MINING

Nama	NIM
Fadia Ramadhana	G64170026

1. Buka data Human_Resource.csv

```
data <- read.csv("Human_Resource.csv", header = TRUE, sep = ",")

str(data)
summary(data)

data$left[data$left==1] <- "Left"
data$left[data$left==0] <- "Stay"
data$left = as.factor(data$left)
levels(data$left)
```

```
> str(data)
'data.frame':   14999 obs. of  10 variables:
 $ satisfaction_level   : num  0.38 0.8 0.11 0.72 0.37 0.41 0.1 0.92 0.89 0.42 ...
 $ last_evaluation      : num  0.53 0.86 0.88 0.87 0.52 0.5 0.77 0.85 1 0.53 ...
 $ number_project       : int  2 5 7 5 2 2 6 5 5 2 ...
 $ average_montly_hours : int  157 262 272 223 159 153 247 259 224 142 ...
 $ time_spend_company   : int  3 6 4 5 3 3 4 5 5 3 ...
 $ work_accident        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ left                 : Factor w/ 2 levels "Left","Stay": 1 1 1 1 1 1 1 1 1 1 ...
 $ promotion_last_5years: int  0 0 0 0 0 0 0 0 0 0 ...
 $ sales                : Factor w/ 10 levels "accounting","hr",..: 8 8 8 8 8 8 8 8 8 8 ...
 $ salary               : Factor w/ 3 levels "high","low","medium": 2 3 3 2 2 2 2 2 2 2 ...

> summary(data)
 satisfaction_level last_evaluation  number_project  average_montly_hours time_spend_company
 Min.   :0.0900     Min.   :0.3600   Min.   :2.000   Min.   : 96.0        Min.   : 2.000
 1st Qu.:0.4400     1st Qu.:0.5600   1st Qu.:3.000   1st Qu.:156.0        1st Qu.: 3.000
 Median :0.6400     Median :0.7200   Median :4.000   Median :200.0        Median : 3.000
 Mean   :0.6128     Mean   :0.7161   Mean   :3.803   Mean   :201.1        Mean   : 3.498
 3rd Qu.:0.8200     3rd Qu.:0.8700   3rd Qu.:5.000   3rd Qu.:245.0        3rd Qu.: 4.000
 Max.   :1.0000     Max.   :1.0000   Max.   :7.000   Max.   :310.0        Max.   :10.000

 Work_accident      left       promotion_last_5years         sales         salary
 Min.   :0.0000   Left: 3571   Min.   :0.00000       sales      :4140   high  :1237
 1st Qu.:0.0000   Stay:11428   1st Qu.:0.00000       technical  :2720   low   :7316
 Median :0.0000                Median :0.00000       support    :2229   medium:6446
 Mean   :0.1446                Mean   :0.02127       IT         :1227
 3rd Qu.:0.0000                3rd Qu.:0.00000       product_mng: 902
 Max.   :1.0000                Max.   :1.00000       marketing  : 858
                                                     (Other)    :2923
```

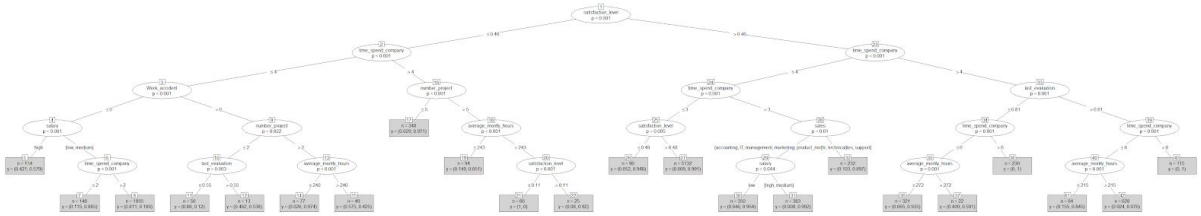
2. Lakukanlah klasifikasi dengan menggunakan fungsi ctree().
    a. Lakukan pengacakan sampel data dan bagi menjadi 70% train dan 30% test

```r
#2
set.seed(1234)

sampel <- sample(2, nrow(data), replace=TRUE, prob=c(0.7,0.3))

trainData <- data[sampel==1,]
testData <- data[sampel==2,]
```

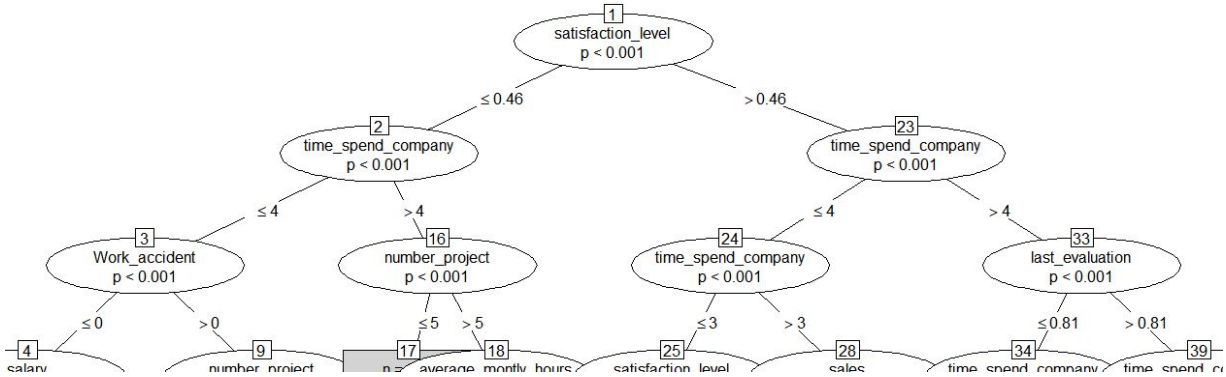
b. Jelaskan hasil dari tree yang terbentuk

```r
myFormula <- left ~ satisfaction_level + last_evaluation + number_project + average_montly_hou
  time_spend_company + Work_accident + salary + promotion_last_5years + sales

hr_ctree <- ctree(myFormula, data = trainData,
                  controls = ctree_control(minsplit = 20, maxdepth = 5))
print(hr_ctree)
plot(hr_ctree)
plot(hr_ctree, type="simple")
```

Tree yang terbentuk :

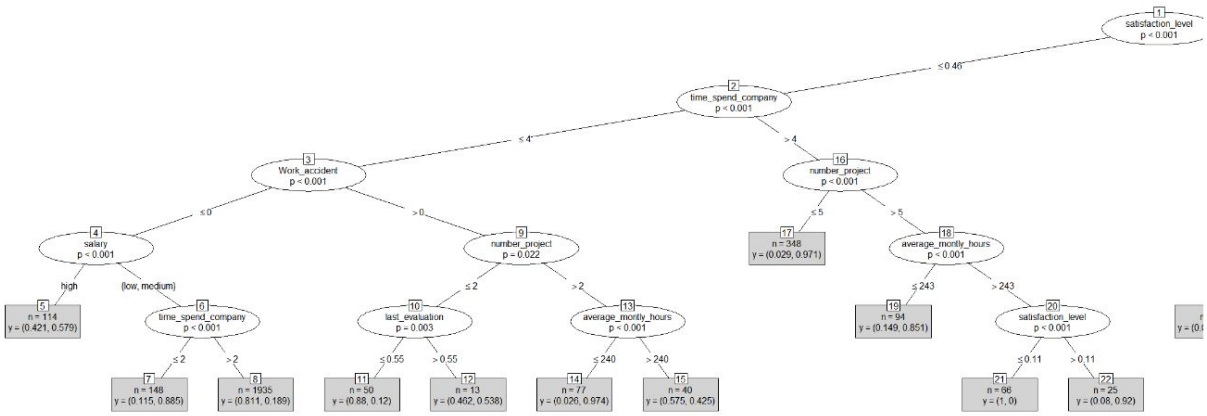


Penjelasan :



Pohon inferensi bersyarat (ctree) menggunakan metode uji signifikansi untuk memilih dan membagi secara rekursif variabel prediktor yang paling terkait dengan hasil. p-value menunjukkan hubungan antara variabel prediktor yang diberikan dengan variabel hasil. Misalnya, node 1 (root) di atas menunjukkan bahwa satisfaction_level adalah variabel yang paling kuat terkait dengan kelas left dengan nilai p < 0,001, dan dengan demikian dipilih sebagai simpul pertama. Selanjutnya node tersebut di split menjadi dua yaitu yang memiliki tingkat kepuasan bekerja (satisfaction_level) dengan value ≤ 0,46 dan yang memiliki tingkat

kepuasaan bekerja (satisfaction_level) dengan value > 0,46. Kemudian dilakukan kembali uji signifikan hingga diperoleh variabel yang paling kuat terkait dengan atribut satisfaction_level untuk kedua jenis value (value ≤ 0,46 dan > 0,46) dan kelas left adalah variabel time_spend_company dengan nilai p-value < 0,001. Ctree menerapkan konsep depth-first search dimana akan membangun tree hingga mencapai leaf pada kedalaman yang telah ditentukan (dalam hal ini maxdepth = 5) terlebih dahulu. Jadi, dimulai dari node root, akan dilakukan uji signifikan untuk satistifaction_level ≤ 0.46 terlebih dahulu dan akan secara rekursif melakukan split pada atribut-atribut yang terpilih selanjutnya hingga mencapai depth = 5. Kemudian akan dilanjutkan uji signifikan untuk satisfaction_level > 0.46 dan secara rekursif melakukan split pada atribut-atribut yang terpilih selanjutnya hingga mencapai depth yang sama. Pada kedalaman tersebut, terdapat node-node leaf. Dan semua node yang berada diantara node root dan node-node leaf disebut dengan decision node.



Node-node yang berwarna abu-abu merupakan node leaf. Sebagai contoh dapat dilihat node ke-5 merupakan node leaf dengan rules :

"*Karyawan yang memiliki tingkat kepuasan bekerja (**satisfaction_level**) ≤ 0,46 lama bekerja di perusahaan (**time_spend_company**) ≤ 4 tahun dan tidak memiliki pengalaman dalam kecelakaan selama bekerja (**Work_accident**) serta tingkat gaji (**salary**) nya termasuk ke dalam kategori high.*"

Jumlah karyawan yang termasuk ke dalam node leaf tersebut ada sebanyak 114 orang dengan proporsi nilai y = (0.421, 0.579) dimana nilai proporsi ini menunjukkan dari 114 orang, 42.1% adalah karyawan yang akan meninggalkan perusahaan dan 57.9% adalah karyawan yang tidak meninggalkan perusahaan.

c. Lakukan prediksi data test dan bandingkan dengan kelas data test awal

```
ctree_pred <- predict(hr_ctree, newdata = testData)
cm <- confusionMatrix(ctree_pred, testData$left)
print(cm)
```

Untuk melakukan prediksi dapat digunakan dengan fungsi predict(). Fungsi

tersebut mengembalikan vektor respons yang diprediksi dari objek tree yang dipasang. **hr_ctree** memiliki peran sebagai object yaitu asumsi sebagai hasil dari beberapa fungsi yang menghasilkan objek dengan komponen berkelas sama seperti yang dikembalikan oleh fungsi pohon. **testData** memiliki peran sebagai newdata yaitu data frame yang berisi nilai-nilai di mana prediksi diperlukan. Kemudian untuk melakukan perbandingan dapat dilakukan dengan menggunakan confusion matrix.

d. Jelaskan hasil confusion matrix

```
Confusion Matrix and Statistics

          Reference
Prediction Left Stay
      Left  962  216
      Stay   93 3225

               Accuracy : 0.9313
                 95% CI : (0.9235, 0.9385)
    No Information Rate : 0.7653
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8161

 Mcnemar's Test P-Value : 3.912e-12

            Sensitivity : 0.9118
            Specificity : 0.9372
         Pos Pred Value : 0.8166
         Neg Pred Value : 0.9720
             Prevalence : 0.2347
         Detection Rate : 0.2140
   Detection Prevalence : 0.2620
      Balanced Accuracy : 0.9245

       'Positive' Class : Left
```

Dari hasil confusion matrix dapat dilihat perbandingan klasifikasi antara tree yang telah dibangun dengan kelas pada data test awal. Data test berjumlah 30% dari total data yaitu 4496 data. Dari matrix, ada sebanyak 1178 karyawan yang diprediksi meninggalkan perusahaan dan ada sebanyak 3318 karyawan yang diprediksi tidak meninggalkan perusahaan. Dari 1178 karyawan, 962 karyawan secara prediksi dan aktual meninggalkan perusahan (TP) dan 261 karyawan secara prediksi meninggalkan perusahaan namun secara aktual tidak meninggalkan perusahaan (FP). Dari 3318 karyawan, 93 karyawan secara prediksi tidak meninggalkan perusahaan namun secara aktual meninggalkan perusahaan (FN) dan 3225 karyawan secara prediksi dan aktual tidak meninggalkan perusahaan (TN).

Dari hasil prediksi tersebut, diperoleh akurasi sebesar 93.13%. Sensitivity adalah ketika secara aktual "meninggalkan perusahaan" seberapa sering tree tersebut memprediksi "meninggalkan perusahaan". Dan nilai Sensitivity yang didapat dari hasil confusion matrix adalah sebesar 91.18%. Specificity mengukur tingkat True Negative yaitu ketika secara aktual "Tidak meninggalkan perusahaan", seberapa sering tree tersebut memprediksi "Tidak meninggalkan perusahaan". Dan nilai Specificity yang didapat dari hasil confusion matrix adalah sebesar 93.72%. Nilai statistik Kappa pada metode ini adalah sebesar 0.8161

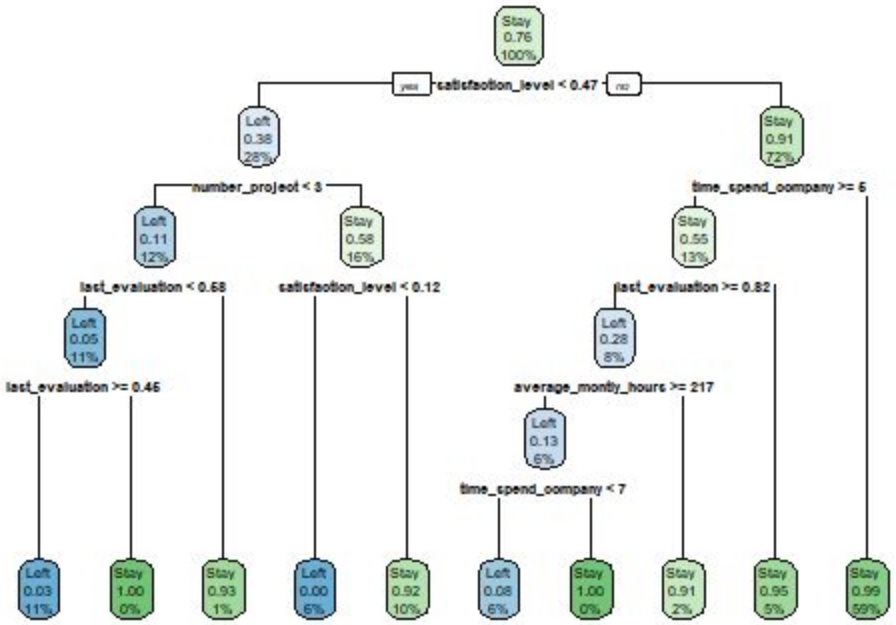
3. Lakukanlah klasifikasi dengan menggunakan fungsi rpart().
   a. Lakukan pengacakan sampel data dan bagi menjadi 70% train dan 30% test

```
#2
set.seed(1234)

sampel <- sample(2, nrow(data), replace=TRUE, prob=c(0.7,0.3))

trainData <- data[sampel==1,]
testData <- data[sampel==2,]
```

   b. Jelaskan hasil dari tree yang terbentuk

```
myFormula <- left ~ satisfaction_level + last_evaluation + number_project + average_montly_hou
  time_spend_company + Work_accident + salary + promotion_last_5years + sales

hr_rpart <- rpart(myFormula, data = trainData, control
                  = rpart.control(minsplit = 10))
print(hr_rpart)
rpart.plot(hr_rpart)
```

Tree yang terbentuk :



Penjelasan :

Dapat dilihat pada tree yang terbentuk bahwa depth terdalam adalah 4.
Pembentukan tree akan dimulai dari root node (depth 0, bagian paling atas dari tree) :

1. Pada root node, dapat dilihat probabilitas keseluruhan untuk tidak meninggalkan perusahaan. Ini menunjukkan proporsi karyawan yang tidak meninggalkan perusahaan. Sebanyak 76% karyawan tidak meninggalkan perusahaan.
2. Root node tersebut menanyakan apakah tingkat kepuasan bekerja (satisfaction_level) karyawan tersebut lebih kecil dari 0.47. Jika “yes”, penelusuran dapat dilakukan ke bawah ke bagian left child node dari root node. 28% merupakan karyawan yang memiliki tingkat kepuasan bekerja (satisfaction_level) lebih kecil dari 0.47 dengan probabilitas 38% dari karyawan tersebut tidak meninggalkan perusahaan.
3. Selanjutnya diberi pertanyaan apakah jumlah project yang telah dikerjakan berjumlah lebih kecil dari 3. Jika “yes”, penelusuran dapat dilakukan ke bawah ke bagian left child node dari node kedua. 12% merupakan karyawan yang telah mengerjakan project kurang dari 3 dengan probabilitas 11% dari karyawan tersebut tidak meninggalkan perusahaan.
4. Selanjutnya diberi pertanyaan apakah nilai evaluasi tahunan (last_evaluation) lebih kecil dari 0.68. Jika “yes”, penelusuran dapat dilakukan ke bawah ke bagian left child node dari root node ketiga. 11%

merupakan karyawan yang nilai evaluasi tahunannya (last_evaluation) lebih kecil dari 0.68 dengan probabilitas 5% dari karyawan tersebut tidak meninggalkan perusahaan.

5. Selanjutnya diberi pertanyaan apakah nilai evaluasi tahunan (last_evaluation) lebih besar atau sama dengan 0.46. Jika "yes", penelusuran dapat dilakukan ke bawah ke bagian left child node dari root node keempat. Node ini merupakan leaf node. 11% merupakan karyawan yang nilai evaluasi tahunannya (last_evaluation) lebih besar dari 0.46 dengan probabilitas 3% dari karyawan tersebut tidak meninggalkan perusahaan.
6. Dan penelusuran tersebut terus dilakukan secara rekursif seperti itu untuk memahami fitur apa yang mempengaruhi kemungkinan karyawan dari perusahaan tersebut akan meninggalkan perusahaan atau tidak.

Secara default, fungsi rpart() menggunakan perhitungan Gini impurity untuk melakukan split node. Semakin tinggi koefisien Gini, semakin beragam instance yang ada dalam node.

c. Lakukan prediksi data test dan bandingkan dengan kelas data test awal

```
rpart_pred <- predict(hr_rpart, newdata = testData, type="class")
cm_rpart <- confusionMatrix(rpart_pred, testData$left)
print(cm_rpart)
```

Untuk melakukan prediksi dapat digunakan dengan fungsi predict(). Fungsi tersebut mengembalikan vektor respons yang diprediksi dari objek tree yang dipasang. **hr_part** memiliki peran sebagai object yaitu asumsi sebagai hasil dari beberapa fungsi yang menghasilkan objek dengan komponen berkelas sama seperti yang dikembalikan oleh fungsi pohon. **testData** memiliki peran sebagai newdata yaitu data frame yang berisi nilai-nilai di mana prediksi diperlukan. Kemudian untuk melakukan perbandingan dapat dilakukan dengan menggunakan confusion matrix.

d. Jelaskan hasil dari confusion matrix

```
Confusion Matrix and Statistics

          Reference
Prediction Left Stay
      Left  968   44
      Stay   87 3397

               Accuracy : 0.9709
                 95% CI : (0.9655, 0.9756)
    No Information Rate : 0.7653
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9177

 Mcnemar's Test P-Value : 0.000243

            Sensitivity : 0.9175
            Specificity : 0.9872
         Pos Pred Value : 0.9565
         Neg Pred Value : 0.9750
             Prevalence : 0.2347
         Detection Rate : 0.2153
   Detection Prevalence : 0.2251
      Balanced Accuracy : 0.9524

       'Positive' Class : Left
```

Penjelasan :

Dari hasil confusion matrix dapat dilihat perbandingan klasifikasi antara tree yang telah dibangun dengan kelas pada data test awal. Data test berjumlah 30% dari total data yaitu 4496 data. Dari matrix, ada sebanyak 1012 karyawan yang diprediksi meninggalkan perusahaan dan ada sebanyak 3484 karyawan yang diprediksi tidak meninggalkan perusahaan. Dari 1012 karyawan, 968 karyawan secara prediksi dan aktual meninggalkan perusahan (TP) dan 44 karyawan secara prediksi meninggalkan perusahaan namun secara aktual tidak meninggalkan perusahaan (FP). Dari 3484 karyawan, 87 karyawan secara prediksi tidak meninggalkan perusahaan namun secara aktual meninggalkan perusahaan (FN) dan 3397 karyawan secara prediksi dan aktual tidak meninggalkan perusahaan (TN).

Dari hasil prediksi tersebut, diperoleh akurasi sebesar 97.09%. Sensitivity adalah ketika secara aktual "meninggalkan perusahaan" seberapa sering tree tersebut memprediksi "meninggalkan perusahaan". Dan nilai Sensitivity yang didapat dari hasil confusion matrix adalah sebesar 91.75%. Specificity mengukur tingkat True Negative yaitu ketika secara aktual "Tidak meninggalkan perusahaan", seberapa sering tree tersebut memprediksi "Tidak meninggalkan perusahaan". Dan nilai Specificity yang didapat dari hasil confusion matrix adalah sebesar 98.72%. Nilai statistik Kappa pada metode ini adalah sebesar 0.9177

4. Lakukanlah klasifikasi dengan menggunakan fungsi SVM().

a. Lakukan pengacakan sampel data dan bagi menjadi 70% train dan 30% test

```
#2
set.seed(1234)

sampel <- sample(2, nrow(data), replace=TRUE, prob=c(0.7,0.3))

trainData <- data[sampel==1,]
testData <- data[sampel==2,]
```

b. Lakukan prediksi data test dan bandingkan dengan kelas data test awal

```
myFormula <- left ~ satisfaction_level + last_evaluation + number_project + average_montly_hou
  time_spend_company + Work_accident + salary + promotion_last_5years + sales

svm_model <- svm(myFormula, data = trainData, cost = 100, gamma = 1)
svm_model

svm_pred <- predict(svm_model, testData)
```

Fungsi svm() pada R digunakan untuk melatih support vector machine. Dapat digunakan untuk melakukan regresi dan klasifikasi umum (tipe nu dan epsilon), serta density estimation. Fungsi svm pada variabel svm_model, memiliki formula yang berasal dari fungsi myFormula sebagai model untuk fitting. Kemudian data frame yang digunakan adalah data train, cost adalah biaya kendala dalam formulasi Lagrange dan dalam hal ini diset 100, gamma adalah parameter yang digunakan oleh semua kernel kecuali kernel Linear dan dalam hal ini diset 1.

```
Call:
svm(formula = myFormula, data = trainData, cost = 100, gamma = 1)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  100

Number of Support Vectors:  2593
```

c. Jelaskan hasil dari confusion matrix

```
confusionMatrix(table(svm_pred, testData$left))
```

```
Confusion Matrix and Statistics


svm_pred Left Stay
    Left  991   50
    Stay   64 3391

               Accuracy : 0.9746
                 95% CI : (0.9696, 0.979)
    No Information Rate : 0.7653
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9291

 Mcnemar's Test P-Value : 0.2234

            Sensitivity : 0.9393
            Specificity : 0.9855
         Pos Pred Value : 0.9520
         Neg Pred Value : 0.9815
             Prevalence : 0.2347
         Detection Rate : 0.2204
   Detection Prevalence : 0.2315
      Balanced Accuracy : 0.9624

       'Positive' Class : Left
```

Penjelasan :

Dari hasil confusion matrix dapat dilihat perbandingan klasifikasi antara tree yang telah dibangun dengan kelas pada data test awal. Data test berjumlah 30% dari total data yaitu 4496 data. Dari matrix, ada sebanyak 1041 karyawan yang diprediksi meninggalkan perusahaan dan ada sebanyak 3455 karyawan yang diprediksi tidak meninggalkan perusahaan. Dari 1041 karyawan, 991 karyawan secara prediksi dan aktual meninggalkan perusahan (TP) dan 50 karyawan secara prediksi meninggalkan perusahaan namun secara aktual tidak meninggalkan perusahaan (FP). Dari 3455 karyawan, 64 karyawan secara prediksi tidak meninggalkan perusahaan namun secara aktual meninggalkan perusahaan (FN) dan 3391 karyawan secara prediksi dan aktual tidak meninggalkan perusahaan (TN).

Dari hasil prediksi tersebut, diperoleh akurasi sebesar 97.46%. Sensitivity adalah ketika secara aktual "meninggalkan perusahaan" seberapa sering tree tersebut memprediksi "meninggalkan perusahaan". Dan nilai Sensitivity yang didapat dari hasil confusion matrix adalah sebesar 93.93%. Specificity mengukur tingkat True Negative yaitu ketika secara aktual "Tidak meninggalkan perusahaan", seberapa sering tree tersebut memprediksi "Tidak meninggalkan perusahaan". Dan nilai Specificity yang didapat dari hasil confusion matrix adalah sebesar 98.55%. Nilai statistik Kappa pada metode ini adalah sebesar 0.9291.

5. Jelaskan perbedaan ketiga hasil klasifikasi tersebut!

Metode	Accuracy	Sensitivity	Specificity	Kappa
ctree()	93.13 %	91.18 %	93.72 %	0.8161
rpart()	97.09 %	91.75 %	98.72 %	0.9177
svm()	97.46 %	93.93%	98.55%	0.9291

Decision tree seperti ctree() dan rpart() adalah algoritma klasifikasi supervised yang berguna ketika variabel input berinteraksi dengan output dengan rule "jika-maka". Mereka juga cocok ketika input memiliki hubungan AND satu sama lain atau ketika variabel input redundan atau berkorelasi. Support Vector Machine (SVM) berguna ketika ada sangat banyak variabel input atau ketika variabel input berinteraksi dengan hasil atau dengan satu sama lain dengan cara yang rumit (nonlinier). Dengan mengamati plot, kita dapat dengan jelas melihat bahwa beberapa variabel saling tidak linier satu sama lain. Oleh karena itu, menggunakan SVM adalah pilihan yang baik pada dataset Human_Resources.csv.

Dari hasil klasifikasi yang didapat, metode rpart() dan svm() dapat mengklasifikasikan data lebih baik dibandingkan dengan ctree(), ini dapat dilihat dari nilai akurasi rpart() sebesar 97.09% dan svm() sebesar 97.46% sedangkan ctree() hanya sebesar 93.13% dan begitu juga dengan nilai-nilai Sensitivity, Specificity, dan Kappa yang lebih besar dibandingkan ctree(). Dari ketiga metode, hasil klasifikasi yang diberikan svm() paling baik untuk dataset Human_Resource.csv. Namun, pada nilai Specificity rpart() memiliki nilai yang lebih unggul dibandingkan dengan svm(). Ini dapat dilihat dari banyaknya data test yang termasuk kedalam True Negative (TN) pada rpart() yaitu 3397 data. Sedangkan banyaknya data test yang termasuk kedalam True Negative (TN) pada svm() yaitu 3391 data.