Report_Final_Project

Ricardo Arturo Figueroa -Juan Diego Fonseca y Angui Cárdenas

2023-11-22

library(tidyverse) library(caret) library(class) library(gmodels) library(DynamicCancerDriverKM) library(rpart) library(randomForest) library(e1071) library(kernlab)

```
normaldata <-(DynamicCancerDriverKM::BRCA_normal)
dataPt <-(DynamicCancerDriverKM::BRCA_PT)
final_data <- bind_rows(normaldata, dataPt)

rate_min_10 <- final_data %>%
               summarise_all(~ mean(. <400, na.rm = TRUE))

colmn_delate <- names(rate_min_10[, rate_min_10 >= 0.8])
data_filter <- final_data %>%
               select(-one_of(colmn_delate))
```

colocar intro

```
print(data_PPInR %>% head(10))
```

```
## # A tibble: 10 x 2
## # Groups:   gen [10]
##    gen    total_mode
##    <chr>       <int>
##  1 TP53          299
##  2 CREBBP        273
##  3 EP300         270
##  4 YWHAG         252
##  5 SMAD3         225
##  6 GRB2          210
##  7 SRC           195
##  8 AR            179
##  9 ESR1          174
## 10 RB1           169
```

k-NN model

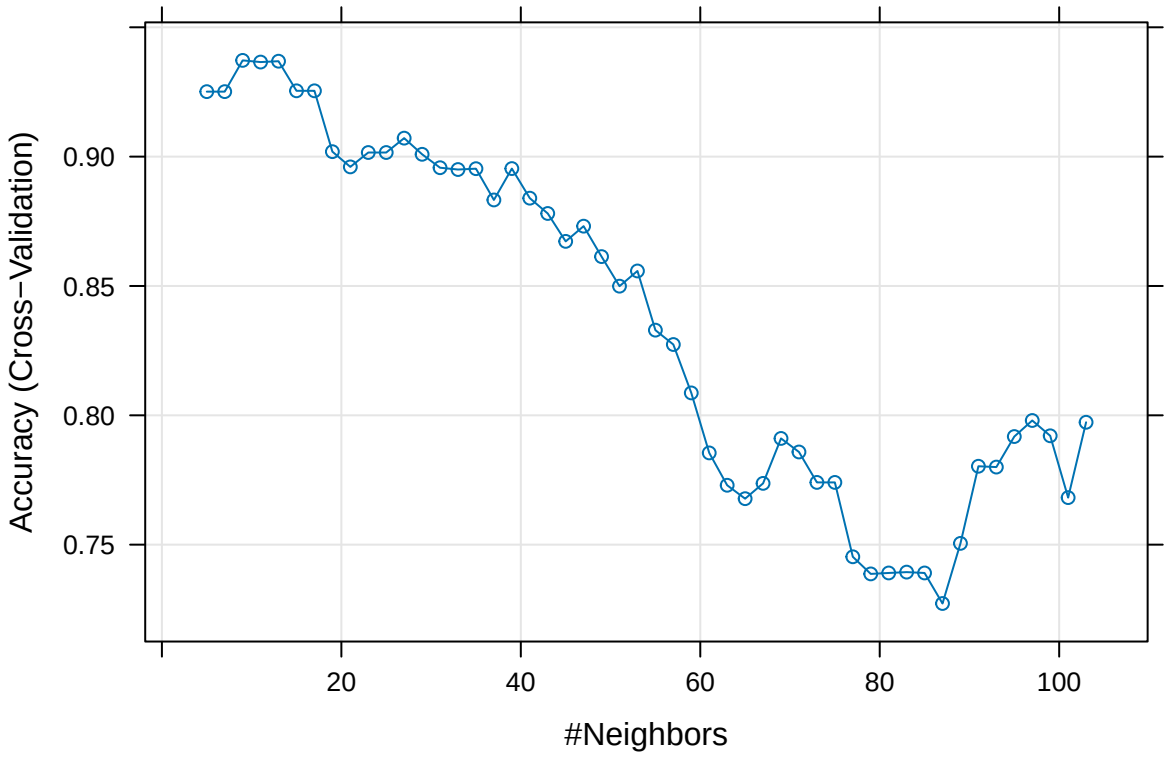
explicar model en Knn

Train the k-NN model

entrenamiento parte 6 a 4

Plot k-NN model

```
plot(knnFit)
```




```
knnPredict <- predict(knnFit, newdata = test.data)
```

Create the confusion matrix for k-NN

cambio de 88 a 90.02 porcianto con la repoarticion del datasets

Linear regression

Summary of linear regression model

explicarlos

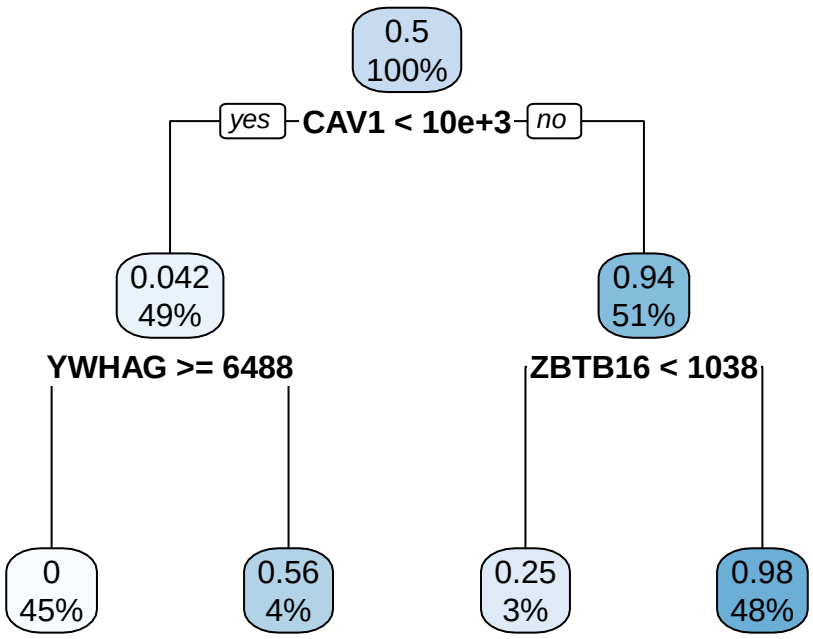
Summarize the results of linear regression model

```
print(model)
```

```
## Linear Regression
##
## 172 samples
## 100 predictors
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 155, 155, 155, 154, 155, 155, ...
## Resampling results:
##
##   RMSE       Rsquared   MAE
##   0.3686969  0.6975828  0.2570416
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

Plot the decision tree

```
rpart.plot::rpart.plot(fit)
```



First Random Forest

```
fit.rf <- randomForest(sample_type ~ .,
                       data = data2_filter[, c(Predictors, "sample_type")])
```

```
## Warning in randomForest.default(m, y, ...): The response has five or fewer
## unique values.  Are you sure you want to do regression?
```

```
prediction.rf <- predict(fit.rf, test.data)
table(test.data$sample_type, prediction.rf)
```

```
##    prediction.rf
##      0 0.002 0.00266666666666667 0.0029 0.0033 0.0039 0.004 0.006 0.0061
##   0  3     4                   1      1      1      1     3     1      1
##   1  0     0                   0      0      0      0     0     0      0
##    prediction.rf
##     0.00613333333333333 0.008 0.0092 0.01 0.0103333333333333 0.0108333333333333
##   0                   1     1      1    1                  1                   1
##   1                   0     0      0    0                  0                   0
##    prediction.rf
##     0.012 0.0135 0.0158333333333333 0.0166666666666667 0.0212 0.024
##   0     1      1                  1                  1      1     1
##   1     0      0                  0                  0      0     0
##    prediction.rf
##     0.0241666666666667 0.0263333333333333 0.0326 0.0354333333333333 0.0438
##   0                  1                  1      1                  1      1
##   1                  0                  0      0                  0      0
##    prediction.rf
##     0.0565666666666667 0.0580666666666667 0.0758 0.0760333333333333 0.1075
##   0                  1                  1      1                  1      1
##   1                  0                  0      0                  0      0
##    prediction.rf
##     0.116233333333333 0.160833333333333 0.7412 0.863433333333333 0.9088
##   0                 1                 1      0                 0      0
##   1                 0                 0      1                 1      1
##    prediction.rf
##     0.931266666666667 0.952166666666667 0.963866666666667 0.964566666666667
##   0                 0                 0                 0                 0
##   1                 1                 1                 2                 1
##    prediction.rf
##     0.977 0.980266666666667 0.982 0.983333333333333 0.9835 0.989633333333333
##   0     0                 0     0                 0      0                 0
##   1     1                 1     1                 1      1                 1
##    prediction.rf
##     0.990533333333333 0.992 0.9956 0.996 0.9987 0.9996  1
##   0                 0     0      0     0      0      0  0
##   1                 1     1      1     3      2      1 11
```

Second Random Forest

```
fit.rf <- randomForest(sample_type ~ .,
                       data = data2_filter[, c(Predictors, "sample_type")])
```

```
## Warning in randomForest.default(m, y, ...): The response has five or fewer
## unique values.  Are you sure you want to do regression?
```

```
prediction.rf <- predict(fit.rf, test.data)
output <- data.frame(Actual = test.data$sample_type, Predicted = prediction.rf)
RMSE = sqrt(sum((output$Actual - output$Predicted)^2) / nrow(output))

print(head(output))
```

```
##   Actual  Predicted
## 1      0 0.00480000
## 2      0 0.00200000
## 3      0 0.01706667
## 4      0 0.00640000
## 5      0 0.02813333
## 6      0 0.04446667
```

######vector

Realiza la búsqueda de hiperparámetros con e1071

```
tune_out <- tune(svm,
                 sample_type ~ .,
                 data = train.data,
                 kernel = "linear",
                 ranges = list(cost = c(0.001, 0.01, 0.1, 1, 5, 10, 100)))
```

Extrae el mejor modelo

```
best_model <- tune_out$best.model
```

Evalúa el rendimiento del modelo

```
library(caret)
confusionMatrix(data = svm_predict, reference = test.data$sample_type)
```

```
## Confusion Matrix and Statistics
##
##           Reference
```

```
## Prediction  0  1
##          0 36  1
##          1  3 34
##
##                Accuracy : 0.9459
##                  95% CI : (0.8673, 0.9851)
##     No Information Rate : 0.527
##     P-Value [Acc > NIR] : 2.071e-15
##
##                   Kappa : 0.8919
##
##  Mcnemar's Test P-Value : 0.6171
##
##             Sensitivity : 0.9231
##             Specificity : 0.9714
##          Pos Pred Value : 0.9730
##          Neg Pred Value : 0.9189
##              Prevalence : 0.5270
##          Detection Rate : 0.4865
##    Detection Prevalence : 0.5000
##       Balanced Accuracy : 0.9473
##
##        'Positive' Class : 0
##
```

##Segunda Parte

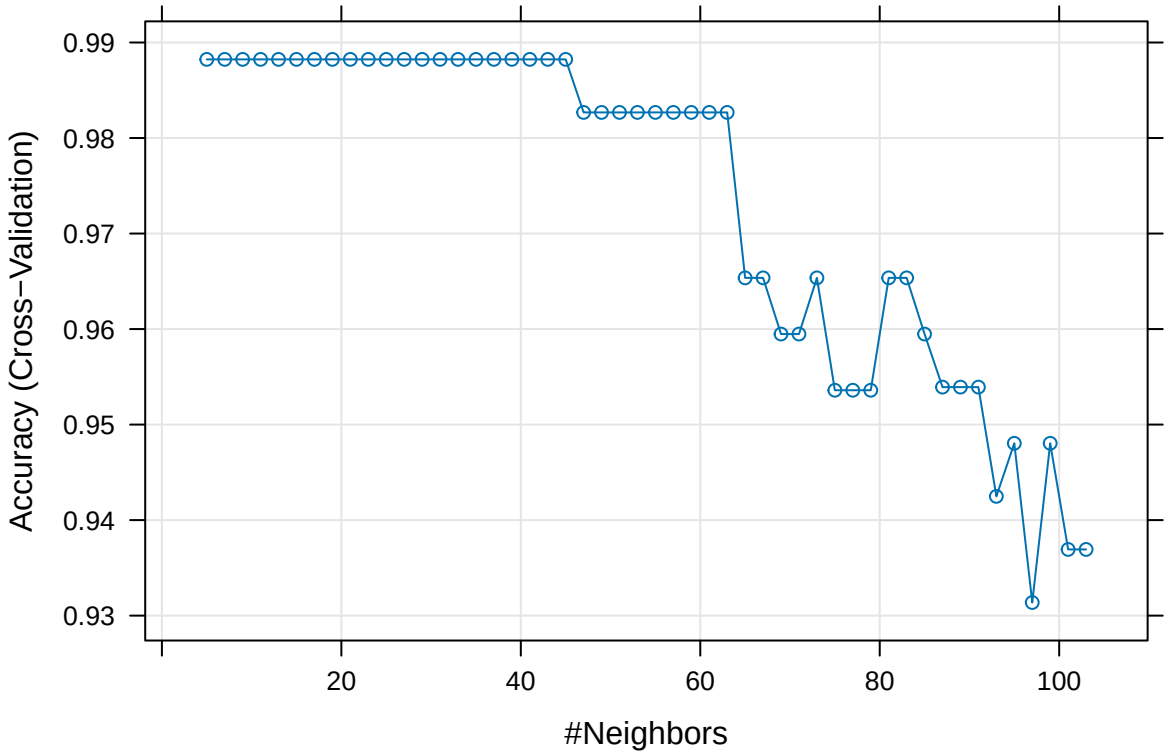
```
folder<-dirname(rstudioapi::getSourceEditorContext()$path)
parentFolder <-dirname(folder)

DataBulks <- file.path(paste0(parentFolder, "/ExperimentsBulk.rdata"))
load(DataBulks)
```

Añadiendo la el DataSet Experiments Bulk

Obtén la columna "features" de gen_scores2

Plot k-NN model

```
plot(knnFit)
```




```
# Make predictions with k-NN
knnPredict <- predict(knnFit, newdata = test.data)
# Create the confusion matrix for k-NN
confusionMatrix(data = knnPredict, reference = test.data$sample_type)
```

```
## Confusion Matrix and Statistics
##
##                      Reference
## Prediction            Primary Tumor Solid Tissue Normal
##   Primary Tumor                  34                   0
##   Solid Tissue Normal             1                  39
##
##                Accuracy : 0.9865
##                  95% CI : (0.927, 0.9997)
##     No Information Rate : 0.527
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9729
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9714
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.9750
##              Prevalence : 0.4730
```

```
##          Detection Rate : 0.4595
##    Detection Prevalence : 0.4595
##       Balanced Accuracy : 0.9857
##
##        'Positive' Class : Primary Tumor
##
```

Linear regression

Fit linear regression model

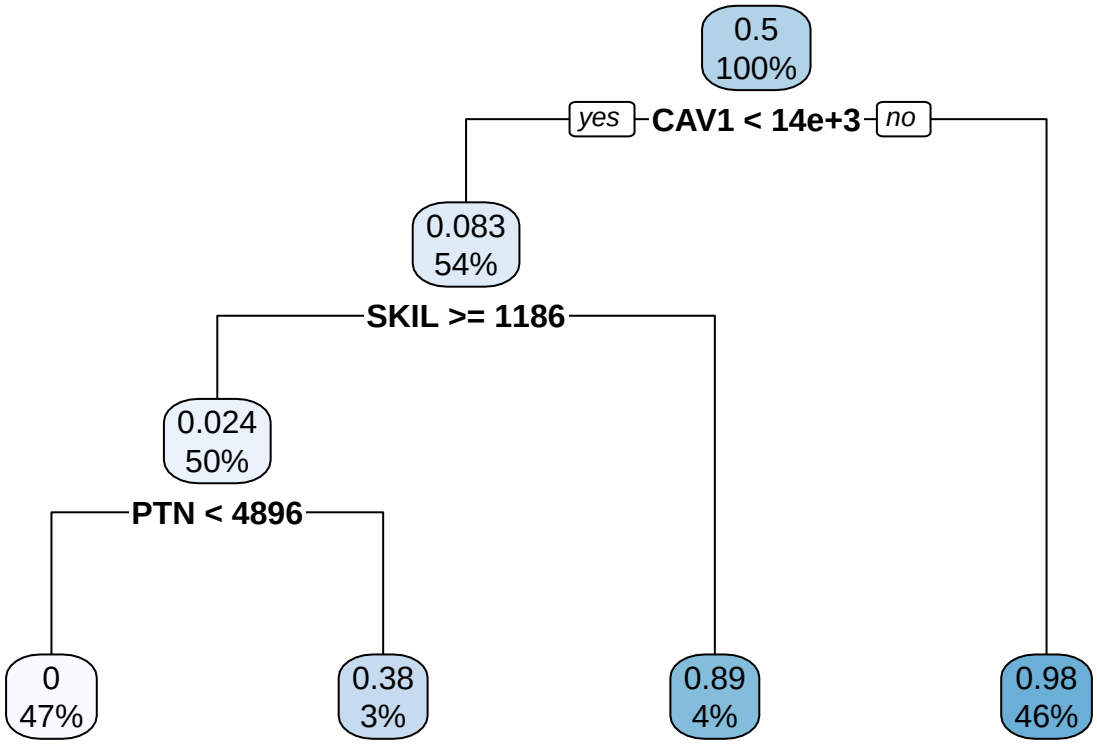
Summarize the results of linear regression model

```
print(model)
```

```
## Linear Regression
##
## 172 samples
## 100 predictors
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 155, 155, 155, 155, 154, 155, ...
## Resampling results:
##
##   RMSE       Rsquared   MAE
##   0.5805922  0.4087009  0.3874165
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

Print the decision tree

```
# Plot the decision tree
rpart.plot::rpart.plot(fit)
```



Bosques aleatorios

```
print(head(output))
```

```
##   Actual  Predicted
## 1      0 0.00000000
## 2      0 0.00200000
## 3      0 0.02600000
## 4      0 0.02076667
## 5      0 0.00850000
## 6      0 0.00640000
```

Realiza predicciones en el conjunto de prueba

```
svm_predict <- predict(svm_model, newdata = test.data)
```

Evalúa el rendimiento del modelo

```
confusionMatrix(data = svm_predict, reference = test.data$sample_type)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 37  1
##          1  0 36
##
##                Accuracy : 0.9865
##                  95% CI : (0.927, 0.9997)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.973
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 1.0000
##             Specificity : 0.9730
##          Pos Pred Value : 0.9737
##          Neg Pred Value : 1.0000
##              Prevalence : 0.5000
##          Detection Rate : 0.5000
##    Detection Prevalence : 0.5135
##       Balanced Accuracy : 0.9865
##
##        'Positive' Class : 0
##
```