Project

2024-07-23

Project

```
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   3.5.1     ✓ 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(dplyr)
library(forcats)
library(ggplot2)
library(rpart)
library(caret)
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(rpart.plot)
library(C50)
library(Metrics)
```

```
##
## Attaching package: 'Metrics'
##
## The following objects are masked from 'package:caret':
##
##     precision, recall
```

```
library(e1071)
library(glmnet)
```

```
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
##
## Loaded glmnet 4.1-8
```

```
options(scipen = 999)

setwd("/Users/19noa/Masters_Program/ADS502/Project")
recruitment <- read.csv("recruitment_data.csv", header = TRUE, sep=',')
head(recruitment,10)
```

```
##    Age Gender EducationLevel ExperienceYears PreviousCompanies
## 1   26      1              2               0                 3
## 2   39      1              4              12                 3
## 3   48      0              2               3                 2
## 4   34      1              2               5                 2
## 5   30      0              1               6                 1
## 6   27      0              3              14                 4
## 7   48      0              2               6                 1
## 8   40      0              4              13                 3
## 9   26      1              3               6                 5
## 10  45      1              2               2                 5
##    DistanceFromCompany InterviewScore SkillScore PersonalityScore
## 1            26.783828             48         78               91
## 2            25.862694             35         68               80
## 3             9.920805             20         67               13
## 4             6.407751             36         27               70
## 5            43.105343             23         52               85
## 6            31.706659             54         50               50
## 7            17.291229             24         52               64
## 8            10.586811              6          3               92
## 9            28.774864             80         78               51
## 10           30.195964             92         16               94
##    RecruitmentStrategy HiringDecision
## 1                    1              1
## 2                    2              1
## 3                    2              0
## 4                    3              0
## 5                    2              0
## 6                    1              1
## 7                    3              0
## 8                    3              0
## 9                    1              1
## 10                   3              0
```

```
recruitment<- recruitment %>%
  mutate(Gender = as.factor(Gender),
         EducationLevel = as.factor(EducationLevel),
         RecruitmentStrategy = as.factor(RecruitmentStrategy),
         HiringDecision = as.factor(HiringDecision),
         Gender_name = as.factor(ifelse(Gender == 0, "Male", "Female")),
         Hiring_name = as.factor(ifelse(HiringDecision == 0, "Not Hired", "Hired")),
         Recruitment_name = fct_collapse(RecruitmentStrategy,
                                         Aggressive = 1,
                                         Moderate = 2,
                                         Conservative = 3),
         Education_name = fct_collapse(EducationLevel,
                                       Bachelor_1 = 1,
                                       Bachelor_2 = 2,
                                       Masters = 3,
                                       PhD = 4))
head(recruitment,10)
```

```
##    Age Gender EducationLevel ExperienceYears PreviousCompanies
## 1   26      1              2               0                 3
## 2   39      1              4              12                 3
## 3   48      0              2               3                 2
## 4   34      1              2               5                 2
## 5   30      0              1               6                 1
## 6   27      0              3              14                 4
## 7   48      0              2               6                 1
## 8   40      0              4              13                 3
## 9   26      1              3               6                 5
## 10  45      1              2               2                 5
##    DistanceFromCompany InterviewScore SkillScore PersonalityScore
## 1            26.783828             48         78               91
## 2            25.862694             35         68               80
## 3             9.920805             20         67               13
## 4             6.407751             36         27               70
## 5            43.105343             23         52               85
## 6            31.706659             54         50               50
## 7            17.291229             24         52               64
## 8            10.586811              6          3               92
## 9            28.774864             80         78               51
## 10           30.195964             92         16               94
##    RecruitmentStrategy HiringDecision Gender_name Hiring_name Recruitment_name
## 1                    1              1      Female       Hired       Aggressive
## 2                    2              1      Female       Hired         Moderate
## 3                    2              0        Male   Not Hired         Moderate
## 4                    3              0      Female   Not Hired     Conservative
## 5                    2              0        Male   Not Hired         Moderate
## 6                    1              1        Male       Hired       Aggressive
## 7                    3              0        Male   Not Hired     Conservative
## 8                    3              0        Male   Not Hired     Conservative
## 9                    1              1      Female       Hired       Aggressive
## 10                   3              0      Female   Not Hired     Conservative
##    Education_name
## 1      Bachelor_2
## 2             PhD
## 3      Bachelor_2
## 4      Bachelor_2
## 5      Bachelor_1
## 6         Masters
## 7      Bachelor_2
## 8             PhD
## 9         Masters
## 10     Bachelor_2
```

```
sapply(recruitment, function(x) sum(is.na(x))) #count total missing values in each column of data frame
```

```
##                 Age              Gender      EducationLevel     ExperienceYears
##                   0                   0                   0                   0
##   PreviousCompanies DistanceFromCompany      InterviewScore          SkillScore
##                   0                   0                   0                   0
##    PersonalityScore RecruitmentStrategy      HiringDecision         Gender_name
##                   0                   0                   0                   0
##         Hiring_name    Recruitment_name      Education_name
##                   0                   0                   0
```

Project

```
summary(recruitment)
```

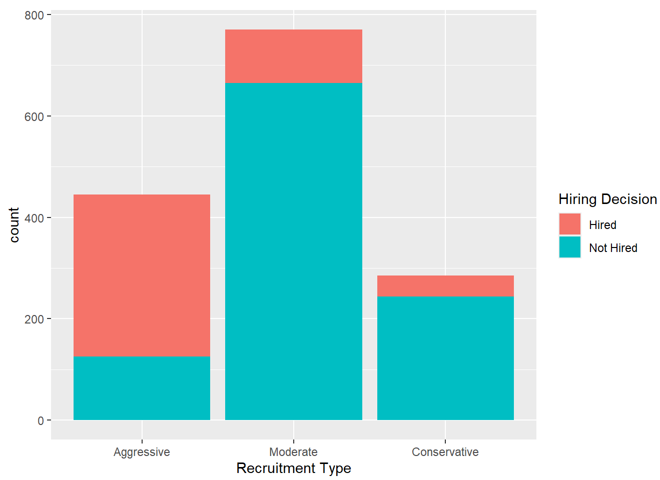
```
##       Age        Gender  EducationLevel ExperienceYears  PreviousCompanies
##  Min.   :20.00   0:762   1:307          Min.   : 0.000   Min.   :1.000
##  1st Qu.:27.00   1:738   2:740          1st Qu.: 4.000   1st Qu.:2.000
##  Median :35.00           3:317          Median : 8.000   Median :3.000
##  Mean   :35.15           4:136          Mean   : 7.694   Mean   :3.002
##  3rd Qu.:43.00                          3rd Qu.:12.000   3rd Qu.:4.000
##  Max.   :50.00                          Max.   :15.000   Max.   :5.000
##  DistanceFromCompany InterviewScore     SkillScore     PersonalityScore
##  Min.   : 1.031      Min.   :  0.00   Min.   :  0.00   Min.   :  0.00
##  1st Qu.:12.839      1st Qu.: 25.00   1st Qu.: 25.75   1st Qu.: 23.00
##  Median :25.502      Median : 52.00   Median : 53.00   Median : 49.00
##  Mean   :25.505      Mean   : 50.56   Mean   : 51.12   Mean   : 49.39
##  3rd Qu.:37.738      3rd Qu.: 75.00   3rd Qu.: 76.00   3rd Qu.: 76.00
##  Max.   :50.992      Max.   :100.00   Max.   :100.00   Max.   :100.00
##  RecruitmentStrategy HiringDecision Gender_name     Hiring_name
##  1:445               0:1035         Female:738   Hired    : 465
##  2:770               1: 465         Male  :762   Not Hired:1035
##  3:285
##
##
##
##      Recruitment_name    Education_name
##  Aggressive  :445     Bachelor_1:307
##  Moderate    :770     Bachelor_2:740
##  Conservative:285     Masters   :317
##                       PhD       :136
##
##
```

```
dim(recruitment)
```

```
## [1] 1500   15
```

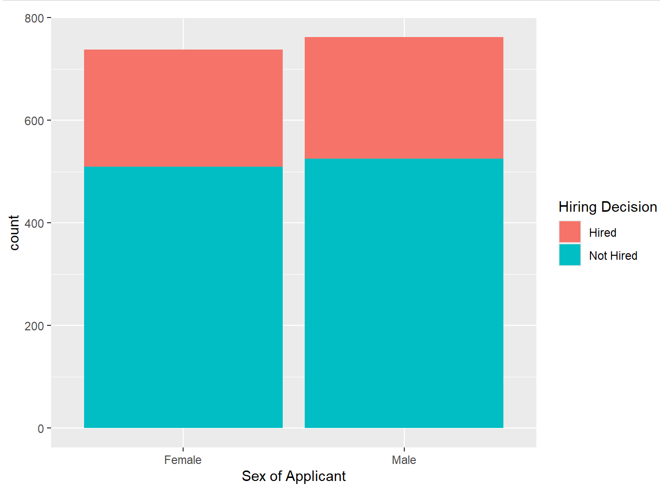
#Project

```
ggplot(data = recruitment) +
geom_bar(aes(x = Recruitment_name, fill = Hiring_name)) +
xlab("Recruitment Type") +
guides(fill = guide_legend(title = "Hiring Decision"))
```



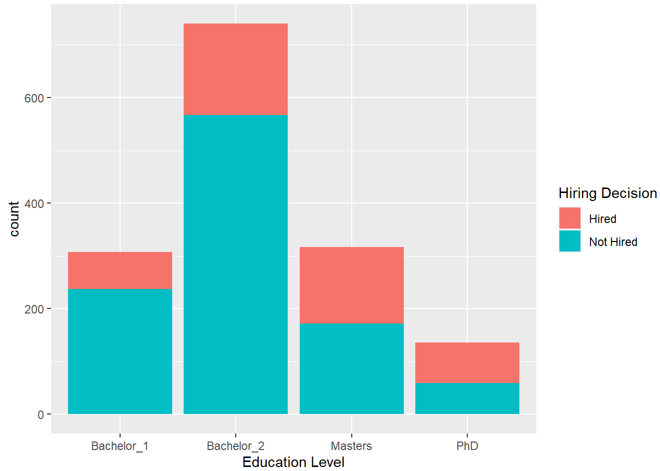
##Project

```
ggplot(data = recruitment) +
geom_bar(aes(x = Gender_name, fill = Hiring_name)) +
xlab("Sex of Applicant") +
guides(fill = guide_legend(title = "Hiring Decision"))
```



Project

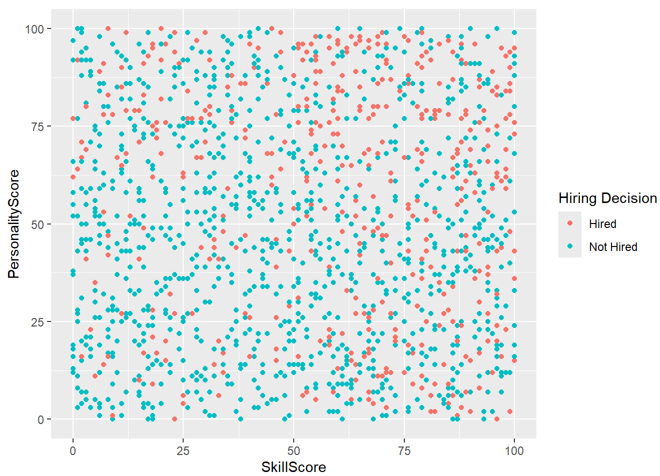
```
ggplot(data = recruitment) +
geom_bar(aes(x = Education_name, fill = Hiring_name)) +
xlab("Education Level") +
guides(fill = guide_legend(title = "Hiring Decision"))
```



##Project

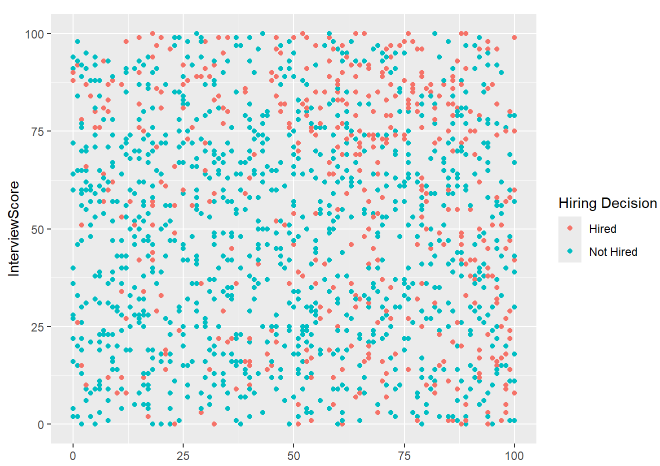
```
hiring_point<- function(x, y){
ggplot(data = recruitment) +
geom_point(aes(x = .data[[x]], y = .data[[y]], color = Hiring_name))
}

hiring_point("SkillScore", "PersonalityScore") +
guides(color = guide_legend(title = "Hiring Decision"))
```



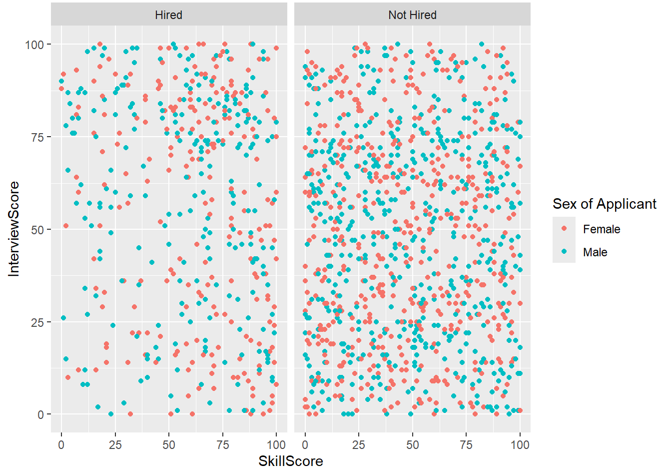
##Project

```
hiring_point("SkillScore", "InterviewScore") +
guides(color = guide_legend(title = "Hiring Decision"))
```

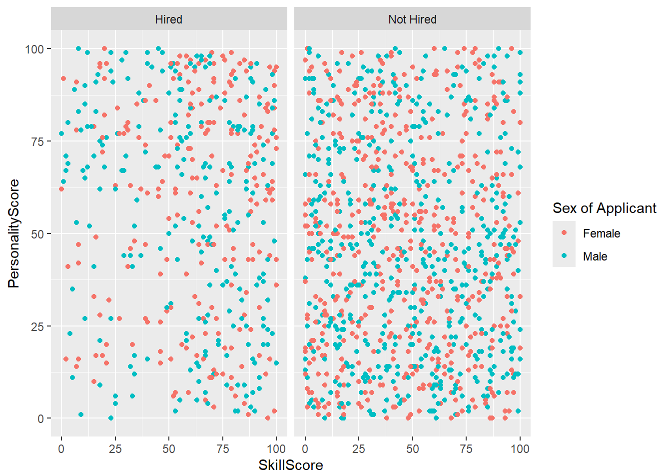


##Project

```
ggplot(data = recruitment) +
geom_point(aes(x = SkillScore, y = InterviewScore, color = Gender_name))+
facet_grid(~Hiring_name) +
guides(color = guide_legend(title = "Sex of Applicant"))
```

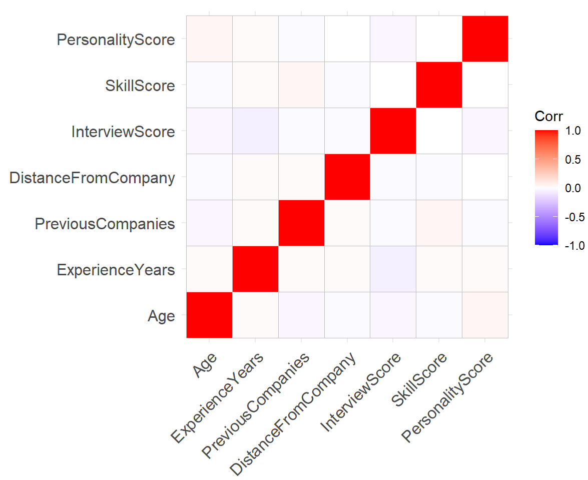



```
ggplot(data = recruitment) +
geom_point(aes(x = SkillScore, y = PersonalityScore, color = Gender_name)) +
facet_grid(~Hiring_name) +
guides(color = guide_legend(title = "Sex of Applicant"))
```




```
#install.packages("ggcorrplot")
#if (!require(devtools)) install.packages("devtools")
#devtools::install_github("kassambara/ggcorrplot")
library(ggcorrplot)
library(ggplot2)
corr<- cor(recruitment[,c(1, 4, 5, 6, 7, 8, 9)])

ggcorrplot(corr)
```



```r
library(caret)
set.seed(720)

trainingRows <- createDataPartition(recruitment$Hiring_name, p = .75, list = FALSE)
recruit_train <- recruitment[trainingRows, ]
recruit_test <- recruitment[-trainingRows, ]

logreg_train <- glm(formula = HiringDecision ~ Age + Gender + EducationLevel + ExperienceYears + PreviousCompanies + Distanc
eFromCompany +InterviewScore + SkillScore + PersonalityScore +
                      RecruitmentStrategy, data = recruit_train, family = binomial)
summary(logreg_train)
```

```
##
## Call:
## glm(formula = HiringDecision ~ Age + Gender + EducationLevel +
##     ExperienceYears + PreviousCompanies + DistanceFromCompany +
##     InterviewScore + SkillScore + PersonalityScore + RecruitmentStrategy,
##     family = binomial, data = recruit_train)
##
## Coefficients:
##                      Estimate Std. Error z value            Pr(>|z|)
## (Intercept)          -4.929373   0.677568  -7.275    0.000000000000346 ***
## Age                  -0.007693   0.010976  -0.701                0.483
## Gender1               0.003851   0.196709   0.020                0.984
## EducationLevel2       0.248679   0.275474   0.903                0.367
## EducationLevel3       2.284181   0.323342   7.064    0.000000000001614 ***
## EducationLevel4       2.605979   0.385030   6.768    0.000000000013034 ***
## ExperienceYears       0.150380   0.023296   6.455    0.000000000108001 ***
## PreviousCompanies     0.098294   0.069243   1.420                0.156
## DistanceFromCompany   0.001221   0.006804   0.180                0.858
## InterviewScore        0.028174   0.003870   7.279    0.000000000000335 ***
## SkillScore            0.032259   0.003700   8.718 < 0.0000000000000002 ***
## PersonalityScore      0.025023   0.003598   6.955    0.000000000003534 ***
## RecruitmentStrategy2 -4.274524   0.276639 -15.452 < 0.0000000000000002 ***
## RecruitmentStrategy3 -4.300213   0.340508 -12.629 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1394.12  on 1125  degrees of freedom
## Residual deviance:  692.71  on 1112  degrees of freedom
## AIC: 720.71
##
## Number of Fisher Scoring iterations: 6
```

```r
logreg_test <- glm(formula = HiringDecision ~ Age + Gender + EducationLevel + ExperienceYears + PreviousCompanies + Distance
FromCompany +InterviewScore + SkillScore + PersonalityScore +
                      RecruitmentStrategy, data = recruit_test, family = binomial)
summary(logreg_test)
```

```
##
## Call:
## glm(formula = HiringDecision ~ Age + Gender + EducationLevel +
##     ExperienceYears + PreviousCompanies + DistanceFromCompany +
##     InterviewScore + SkillScore + PersonalityScore + RecruitmentStrategy,
##     family = binomial, data = recruit_test)
##
## Coefficients:
##                       Estimate Std. Error z value            Pr(>|z|)
## (Intercept)          -5.2596876  1.2418945  -4.235    0.000022833538639 ***
## Age                   0.0002623  0.0177414   0.015             0.988204
## Gender1              -0.4334844  0.3328531  -1.302             0.192804
## EducationLevel2      -0.0806797  0.4550361  -0.177             0.859270
## EducationLevel3       1.8626172  0.5199687   3.582             0.000341 ***
## EducationLevel4       2.6132127  0.6779406   3.855             0.000116 ***
## ExperienceYears       0.1807383  0.0372920   4.847    0.000001256169147 ***
## PreviousCompanies     0.3008125  0.1189031   2.530             0.011410 *
## DistanceFromCompany  -0.0133079  0.0111362  -1.195             0.232079
## InterviewScore        0.0195084  0.0057369   3.401             0.000673 ***
## SkillScore            0.0330248  0.0066788   4.945    0.000000762576832 ***
## PersonalityScore      0.0316919  0.0064969   4.878    0.000001071758140 ***
## RecruitmentStrategy2 -4.0377925  0.4672625  -8.641 < 0.0000000000000002 ***
## RecruitmentStrategy3 -4.1031708  0.5668364  -7.239    0.000000000000453 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 463.18  on 373  degrees of freedom
## Residual deviance: 247.30  on 360  degrees of freedom
## AIC: 275.3
##
## Number of Fisher Scoring iterations: 6
```

```r
##Education Level, Age, Gender, Previous Companies, and Distance from company will be removed since they are not statistical
ly significant (p > .05)and this has been validated with the testing data set.
```

```r
logreg_train <- glm(formula = HiringDecision ~ ExperienceYears +InterviewScore + SkillScore + PersonalityScore + Recruitment
Strategy, data = recruit_train, family = binomial)
summary(logreg_train)
```

```
##
## Call:
## glm(formula = HiringDecision ~ ExperienceYears + InterviewScore +
##     SkillScore + PersonalityScore + RecruitmentStrategy, family = binomial,
##     data = recruit_train)
##
## Coefficients:
##                       Estimate Std. Error z value            Pr(>|z|)
## (Intercept)          -3.561670   0.401681  -8.867 < 0.0000000000000002 ***
## ExperienceYears       0.129854   0.021149   6.140 0.000000000825769362 ***
## InterviewScore        0.024599   0.003469   7.091 0.000000000001335295 ***
## SkillScore            0.027301   0.003326   8.207 0.000000000000000227 ***
## PersonalityScore      0.023968   0.003259   7.356 0.000000000000190030 ***
## RecruitmentStrategy2 -3.716267   0.233168 -15.938 < 0.0000000000000002 ***
## RecruitmentStrategy3 -3.738416   0.288808 -12.944 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1394.12  on 1125  degrees of freedom
## Residual deviance:  812.45  on 1119  degrees of freedom
## AIC: 826.45
##
## Number of Fisher Scoring iterations: 6
```

```
predicted_train <- predict(logreg_train, newdata = recruit_train, type = "response")

predicted_classes <- ifelse(predicted_train > 0.5, 1, 0)

t1 <- table(recruit_train$HiringDecision, predicted_classes)
row.names(t1) <- c("Actual: 0", "Actual: 1")
colnames(t1) <- c("Predicted: 0", "Predicted: 1")
t1 <- addmargins(A = t1, FUN = list(Total = sum), quiet = TRUE)
t1
```

```
##            predicted_classes
##             Predicted: 0 Predicted: 1 Total
##   Actual: 0          717           60   777
##   Actual: 1          106          243   349
##   Total              823          303  1126
```

```
TN <- t1[1,1]
#print(paste("TN:", TN))
TP <- t1[2,2]
#print(paste("TP:", TP))
FN <- t1[2,1]
#print(paste("FN:", FN))
FP <- t1[1,2]
#print(paste("FP:", FP))
GT <- TN + TP + FN + FP

accuracy <- (TP + TN) / GT
error_rate <- 1 - accuracy
sensitivity <- TP / (TP + FN)
specificity <- TN / (TN + FP)
precision <- TP / (TP + FP)
recall <- TP / (TP + FN)
f1 <- 2 * (precision * recall) / (precision + recall)
f2 <- (5 * precision * recall) / ((4 * precision) + recall)
f05 <- ((1.25 * precision * recall) / ((0.25 * precision) + recall))

model_evaluation_table <- data.frame(
  Evaluation_Measure = c("Accuracy", "Error Rate", "Sensitivity", "Specificity", "Precision", "F1 Score", "F2 Score", "F0.5
Score"),
  Logistic_Regression_Training_Model = c(accuracy, error_rate, sensitivity, specificity, precision, f1, f2, f05)
)

model_evaluation_table
```

```
##   Evaluation_Measure Logistic_Regression_Training_Model
## 1           Accuracy                          0.8525755
## 2         Error Rate                          0.1474245
## 3        Sensitivity                          0.6962751
## 4        Specificity                          0.9227799
## 5          Precision                          0.8019802
## 6           F1 Score                          0.7453988
## 7           F2 Score                          0.7151265
## 8         F0.5 Score                          0.7783472
```

```
logreg_test <- glm(formula = HiringDecision ~ ExperienceYears + InterviewScore + SkillScore + PersonalityScore + Recruitment
Strategy, data = recruit_test, family = binomial)
summary(logreg_test)
```

```
##
## Call:
## glm(formula = HiringDecision ~ ExperienceYears + InterviewScore +
##     SkillScore + PersonalityScore + RecruitmentStrategy, family = binomial,
##     data = recruit_test)
##
## Coefficients:
##                       Estimate Std. Error z value            Pr(>|z|)
## (Intercept)          -3.402631   0.622422  -5.467     0.00000004583293 ***
## ExperienceYears       0.165936   0.034035   4.875     0.00000108536366 ***
## InterviewScore        0.016039   0.005102   3.144             0.00167 **
## SkillScore            0.023961   0.005566   4.305     0.00001672708510 ***
## PersonalityScore      0.022160   0.005495   4.033     0.00005513146899 ***
## RecruitmentStrategy2 -3.250618   0.372017  -8.738 < 0.0000000000000002 ***
## RecruitmentStrategy3 -3.121314   0.458295  -6.811     0.00000000000971 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 463.18  on 373  degrees of freedom
## Residual deviance: 294.15  on 367  degrees of freedom
## AIC: 308.15
##
## Number of Fisher Scoring iterations: 5
```

```
predicted_test <- predict(logreg_test, newdata = recruit_test, type = "response")

predicted_test_classes <- ifelse(predicted_test > 0.5, 1, 0)

t2 <- table(recruit_test$HiringDecision, predicted_test_classes)
row.names(t2) <- c("Actual: 0", "Actual: 1")
colnames(t2) <- c("Predicted: 0", "Predicted: 1")
t2 <- addmargins(A = t2, FUN = list(Total = sum), quiet = TRUE)
t2
```

```
##           predicted_test_classes
##            Predicted: 0 Predicted: 1 Total
##   Actual: 0         240           18   258
##   Actual: 1          45           71   116
##   Total             285           89   374
```

```
TN1 <- t2[1,1]
#print(paste("TN:", TN1))
TP1 <- t2[2,2]
#print(paste("TP:", TP1))
FN1 <- t2[2,1]
#print(paste("FN:", FN1))
FP1 <- t2[1,2]
#print(paste("FP:", FP1))
GT1 <- TN1 + TP1 + FN1 + FP1

accuracy1 <- (TP1 + TN1) / GT1
error_rate1 <- 1 - accuracy1
sensitivity1 <- TP1 / (TP1 + FN1)
specificity1 <- TN1 / (TN1 + FP1)
precision1 <- TP1 / (TP1 + FP1)
recall1 <- TP1 / (TP1 + FN1)
f1_1 <- 2 * (precision1 * recall1) / (precision1 + recall1)
f2_1 <- (5 * precision1 * recall1) / ((4 * precision1) + recall1)
f05_1 <- ((1.25 * precision1 * recall1) / ((0.25 * precision1) + recall1))

model_evaluation_table <- data.frame(
  Evaluation_Measure = c("Accuracy", "Error Rate", "Sensitivity", "Specificity", "Precision", "F1 Score", "F2 Score", "F0.5
Score"),
  Logistic_Regression_Training_Model = c(accuracy, error_rate, sensitivity, specificity, precision, f1, f2, f05),
  Logistic_Regression_Testing_Model = c(accuracy1, error_rate1, sensitivity1, specificity1, precision1, f1_1, f2_1, f05_1)
)

model_evaluation_table
```

```
##   Evaluation_Measure Logistic_Regression_Training_Model
## 1           Accuracy                          0.8525755
## 2         Error Rate                          0.1474245
## 3        Sensitivity                          0.6962751
## 4        Specificity                          0.9227799
## 5          Precision                          0.8019802
## 6           F1 Score                          0.7453988
## 7           F2 Score                          0.7151265
## 8         F0.5 Score                          0.7783472
##   Logistic_Regression_Testing_Model
## 1                         0.8315508
## 2                         0.1684492
## 3                         0.6120690
## 4                         0.9302326
## 5                         0.7977528
## 6                         0.6926829
## 7                         0.6419530
## 8                         0.7521186
```

```
nb01 <- naiveBayes(formula = HiringDecision ~ ExperienceYears + InterviewScore + SkillScore + PersonalityScore + Recruitment
Strategy, data = recruit_train)
ypred <- predict(object = nb01, newdata = recruit_train)
nb01
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##         0         1
## 0.6900533 0.3099467
##
## Conditional probabilities:
##    ExperienceYears
## Y       [,1]     [,2]
##   0 7.486486 4.707434
##   1 8.386819 4.346613
##
##    InterviewScore
## Y       [,1]     [,2]
##   0 48.01673 27.22327
##   1 57.73352 29.86551
##
##    SkillScore
## Y       [,1]     [,2]
##   0 47.12355 29.62063
##   1 60.75358 26.82633
##
##    PersonalityScore
## Y       [,1]     [,2]
##   0 45.43243 28.66485
##   1 56.53582 30.27037
##
##    RecruitmentStrategy
## Y            1          2          3
##   0 0.11325611 0.64864865 0.23809524
##   1 0.69054441 0.22349570 0.08595989
```

```
t3 <- table(recruit_train$HiringDecision, ypred)
row.names(t3) <- c("Actual: 0", "Actual: 1")
colnames(t3) <- c("Predicted: 0", "Predicted: 1")
t3 <- addmargins(A = t3, FUN = list(Total = sum), quiet = TRUE)
t3
```

```
##            ypred
##             Predicted: 0 Predicted: 1 Total
##   Actual: 0          721           56   777
##   Actual: 1          119          230   349
##   Total              840          286  1126
```

```
TN2 <- t3[1,1]
TP2 <- t3[2,2]
FN2 <- t3[2,1]
FP2 <- t3[1,2]
GT2 <- TN2 + TP2 + FN2 + FP2

accuracy2 <- (TP2 + TN2) / GT2
error_rate2 <- 1 - accuracy2
sensitivity2 <- TP2 / (TP2 + FN2)
specificity2 <- TN2 / (TN2 + FP2)
precision2 <- TP2 / (TP2 + FP2)
recall2 <- TP2 / (TP2 + FN2)
f1_2 <- 2 * (precision2 * recall2) / (precision2 + recall2)
f2_2 <- (5 * precision2 * recall2) / ((4 * precision2) + recall2)
f05_2 <- ((1.25 * precision2 * recall2) / ((0.25 * precision2) + recall2))

model_evaluation_table <- data.frame(
  Evaluation_Measure = c("Accuracy", "Error Rate", "Sensitivity", "Specificity", "Precision", "F1 Score", "F2 Score", "F0.5
Score"),
  Logistic_Regression_Training_Model = c(accuracy, error_rate, sensitivity, specificity, precision, f1, f2, f05),
  Logistic_Regression_Testing_Model = c(accuracy1, error_rate1, sensitivity1, specificity1, precision1, f1_1, f2_1, f05_1),
  Naive_Bayes_Training_Model = c(accuracy2, error_rate2, sensitivity2, specificity2, precision2, f1_2, f2_2, f05_2)
)

model_evaluation_table
```

```
##   Evaluation_Measure Logistic_Regression_Training_Model
## 1           Accuracy                          0.8525755
## 2         Error Rate                          0.1474245
## 3        Sensitivity                          0.6962751
## 4        Specificity                          0.9227799
## 5          Precision                          0.8019802
## 6           F1 Score                          0.7453988
## 7           F2 Score                          0.7151265
## 8         F0.5 Score                          0.7783472
##   Logistic_Regression_Testing_Model Naive_Bayes_Training_Model
## 1                         0.8315508                  0.8445826
## 2                         0.1684492                  0.1554174
## 3                         0.6120690                  0.6590258
## 4                         0.9302326                  0.9279279
## 5                         0.7977528                  0.8041958
## 6                         0.6926829                  0.7244094
## 7                         0.6419530                  0.6837099
## 8                         0.7521186                  0.7702612
```

```
nb01_test <- naiveBayes(formula = HiringDecision ~ ExperienceYears + InterviewScore + SkillScore + PersonalityScore + Recrui
tmentStrategy, data = recruit_test)
ypred_test <- predict(object = nb01_test, newdata = recruit_test)
nb01_test
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##         0         1
## 0.6898396 0.3101604
##
## Conditional probabilities:
##    ExperienceYears
## Y       [,1]     [,2]
##   0 6.790698 4.733644
##   1 9.008621 4.324844
##
##    InterviewScore
## Y       [,1]     [,2]
##   0 46.99612 28.41090
##   1 53.99138 30.61699
##
##    SkillScore
## Y       [,1]     [,2]
##   0 47.06977 28.31979
##   1 57.86207 29.47037
##
##    PersonalityScore
## Y       [,1]     [,2]
##   0 47.94961 28.30286
##   1 57.56897 28.11598
##
##    RecruitmentStrategy
## Y            1          2          3
##   0 0.14728682 0.62403101 0.22868217
##   1 0.67241379 0.23275862 0.09482759
```

```
t4 <- table(recruit_test$HiringDecision, ypred_test)
row.names(t4) <- c("Actual: 0", "Actual: 1")
colnames(t4) <- c("Predicted: 0", "Predicted: 1")
t4 <- addmargins(A = t4, FUN = list(Total = sum), quiet = TRUE)
t4
```

```
##            ypred_test
##             Predicted: 0 Predicted: 1 Total
##   Actual: 0          239           19   258
##   Actual: 1           45           71   116
##   Total              284           90   374
```

```
TN3 <- t4[1,1]
TP3 <- t4[2,2]
FN3 <- t4[2,1]
FP3 <- t4[1,2]
GT3 <- TN3 + TP3 + FN3 + FP3

accuracy3 <- (TP3 + TN3) / GT3
error_rate3 <- 1 - accuracy3
sensitivity3 <- TP3 / (TP3 + FN3)
specificity3 <- TN3 / (TN3 + FP3)
precision3 <- TP3 / (TP3 + FP3)
recall3 <- TP3 / (TP3 + FN3)
f1_3 <- 2 * (precision3 * recall3) / (precision3 + recall3)
f2_3 <- (5 * precision3 * recall3) / ((4 * precision3) + recall3)
f05_3 <- ((1.25 * precision3 * recall3) / ((0.25 * precision3) + recall3))

model_evaluation_table <- data.frame(
  Evaluation_Measure = c("Accuracy", "Error Rate", "Sensitivity", "Specificity", "Precision", "F1 Score", "F2 Score", "F0.5 Score"),
  Logistic_Regression_Training_Model = c(accuracy, error_rate, sensitivity, specificity, precision, f1, f2, f05),
  Logistic_Regression_Testing_Model = c(accuracy1, error_rate1, sensitivity1, specificity1, precision1, f1_1, f2_1, f05_1),
  Naive_Bayes_Training_Model = c(accuracy2, error_rate2, sensitivity2, specificity2, precision2, f1_2, f2_2, f05_2),
  Naive_Bayes_Testing_Model = c(accuracy3, error_rate3, sensitivity3, specificity3, precision3, f1_3, f2_3, f05_3)
)

model_evaluation_table
```

```
##   Evaluation_Measure Logistic_Regression_Training_Model
## 1           Accuracy                          0.8525755
## 2         Error Rate                          0.1474245
## 3        Sensitivity                          0.6962751
## 4        Specificity                          0.9227799
## 5          Precision                          0.8019802
## 6           F1 Score                          0.7453988
## 7           F2 Score                          0.7151265
## 8         F0.5 Score                          0.7783472
##   Logistic_Regression_Testing_Model Naive_Bayes_Training_Model
## 1                         0.8315508                  0.8445826
## 2                         0.1684492                  0.1554174
## 3                         0.6120690                  0.6590258
## 4                         0.9302326                  0.9279279
## 5                         0.7977528                  0.8041958
## 6                         0.6926829                  0.7244094
## 7                         0.6419530                  0.6837099
## 8                         0.7521186                  0.7702612
##   Naive_Bayes_Testing_Model
## 1                 0.8288770
## 2                 0.1711230
## 3                 0.6120690
## 4                 0.9263566
## 5                 0.7888889
## 6                 0.6893204
## 7                 0.6407942
## 8                 0.7457983
```

```
```