finalProject632

2023-05-01

```
data<-read.csv("/Users/sunny/Downloads/HRDATA.csv")
summary(data)
```

```
##  satisfaction_level last_evaluation  number_project  average_montly_hours
##  Min.   :0.0900     Min.   :0.3600   Min.   :2.000   Min.   : 96.0
##  1st Qu.:0.4400     1st Qu.:0.5600   1st Qu.:3.000   1st Qu.:156.0
##  Median :0.6400     Median :0.7200   Median :4.000   Median :200.0
##  Mean   :0.6128     Mean   :0.7161   Mean   :3.803   Mean   :201.1
##  3rd Qu.:0.8200     3rd Qu.:0.8700   3rd Qu.:5.000   3rd Qu.:245.0
##  Max.   :1.0000     Max.   :1.0000   Max.   :7.000   Max.   :310.0
##  time_spend_company Work_accident         left        promotion_last_5years
##  Min.   : 2.000     Min.   :0.0000   Min.   :0.0000   Min.   :0.00000
##  1st Qu.: 3.000     1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.00000
##  Median : 3.000     Median :0.0000   Median :0.0000   Median :0.00000
##  Mean   : 3.498     Mean   :0.1446   Mean   :0.2381   Mean   :0.02127
##  3rd Qu.: 4.000     3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.00000
##  Max.   :10.000     Max.   :1.0000   Max.   :1.0000   Max.   :1.00000
##   Department           salary
##  Length:14999       Length:14999
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
```

```
data1<-read.csv("/Users/sunny/Downloads/HRDATA.csv")
summary(data1)
```

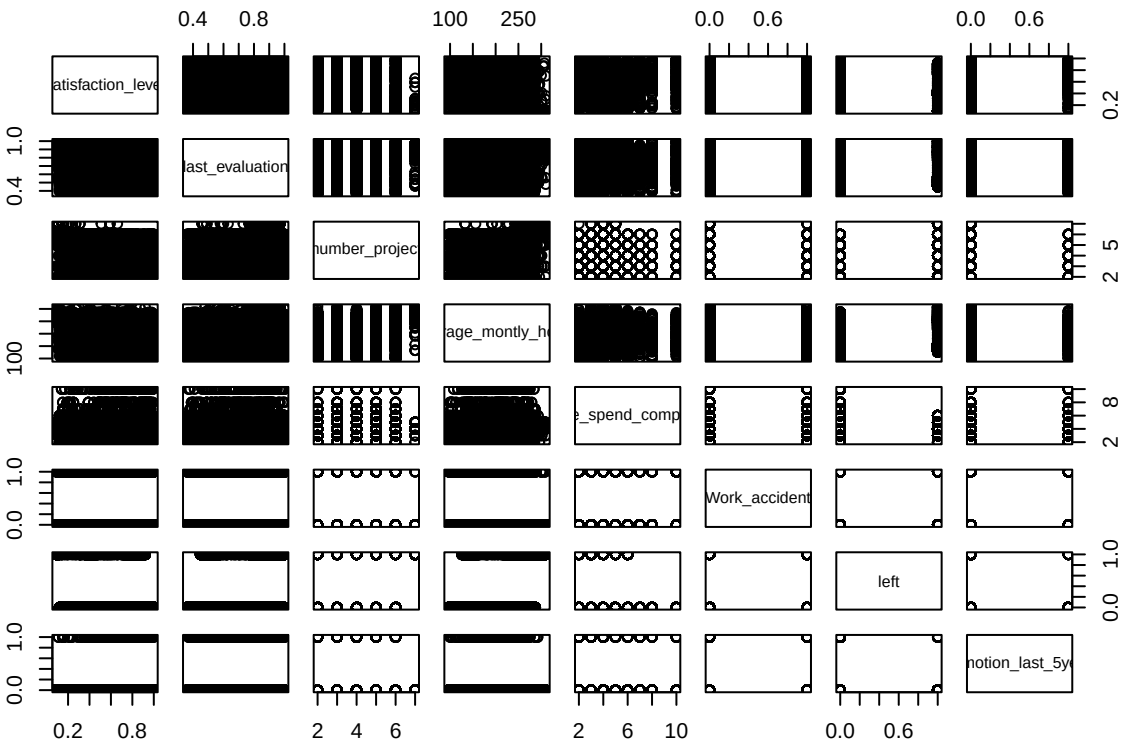
```
##  satisfaction_level last_evaluation  number_project  average_montly_hours
##  Min.   :0.0900     Min.   :0.3600   Min.   :2.000   Min.   : 96.0
##  1st Qu.:0.4400     1st Qu.:0.5600   1st Qu.:3.000   1st Qu.:156.0
##  Median :0.6400     Median :0.7200   Median :4.000   Median :200.0
##  Mean   :0.6128     Mean   :0.7161   Mean   :3.803   Mean   :201.1
##  3rd Qu.:0.8200     3rd Qu.:0.8700   3rd Qu.:5.000   3rd Qu.:245.0
##  Max.   :1.0000     Max.   :1.0000   Max.   :7.000   Max.   :310.0
##  time_spend_company Work_accident         left        promotion_last_5years
##  Min.   : 2.000     Min.   :0.0000   Min.   :0.0000   Min.   :0.00000
##  1st Qu.: 3.000     1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.00000
##  Median : 3.000     Median :0.0000   Median :0.0000   Median :0.00000
##  Mean   : 3.498     Mean   :0.1446   Mean   :0.2381   Mean   :0.02127
##  3rd Qu.: 4.000     3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.00000
##  Max.   :10.000     Max.   :1.0000   Max.   :1.0000   Max.   :1.00000
##   Department           salary
##  Length:14999       Length:14999
##  Class :character   Class :character
##  Mode  :character   Mode  :character
```

```
##
##
##
```

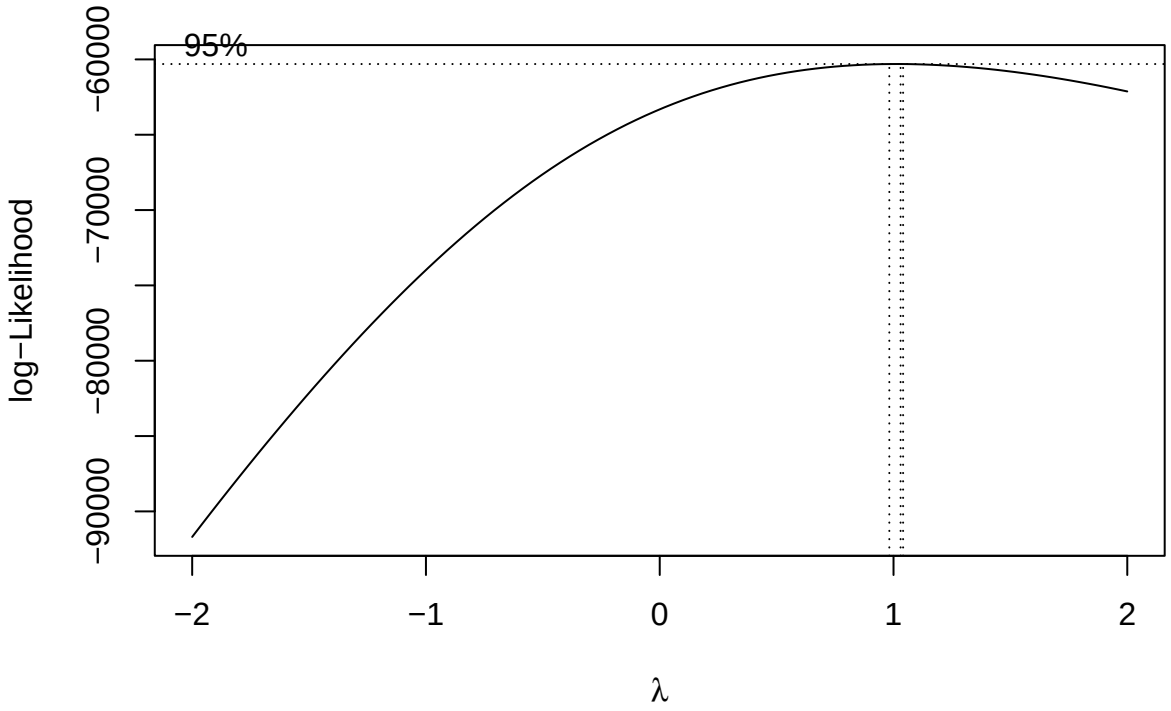
```
head(data1)
```

```
##   satisfaction_level last_evaluation number_project average_montly_hours
## 1               0.38            0.53              2                   157
## 2               0.80            0.86              5                   262
## 3               0.11            0.88              7                   272
## 4               0.72            0.87              5                   223
## 5               0.37            0.52              2                   159
## 6               0.41            0.50              2                   153
##   time_spend_company Work_accident left promotion_last_5years Department salary
## 1                  3             0    1                     0      sales    low
## 2                  6             0    1                     0      sales medium
## 3                  4             0    1                     0      sales medium
## 4                  5             0    1                     0      sales    low
## 5                  3             0    1                     0      sales    low
## 6                  3             0    1                     0      sales    low
```

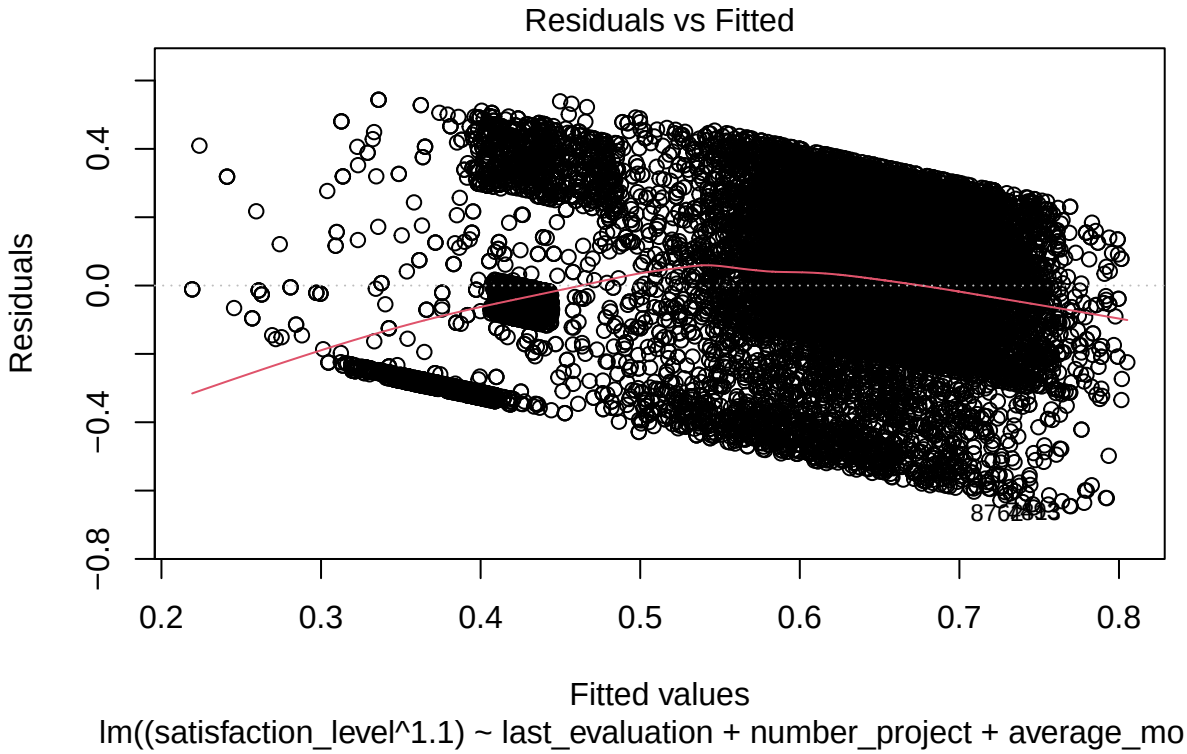
```
pairs(data[c(1,2,3,4,5,6,7,8)])
```

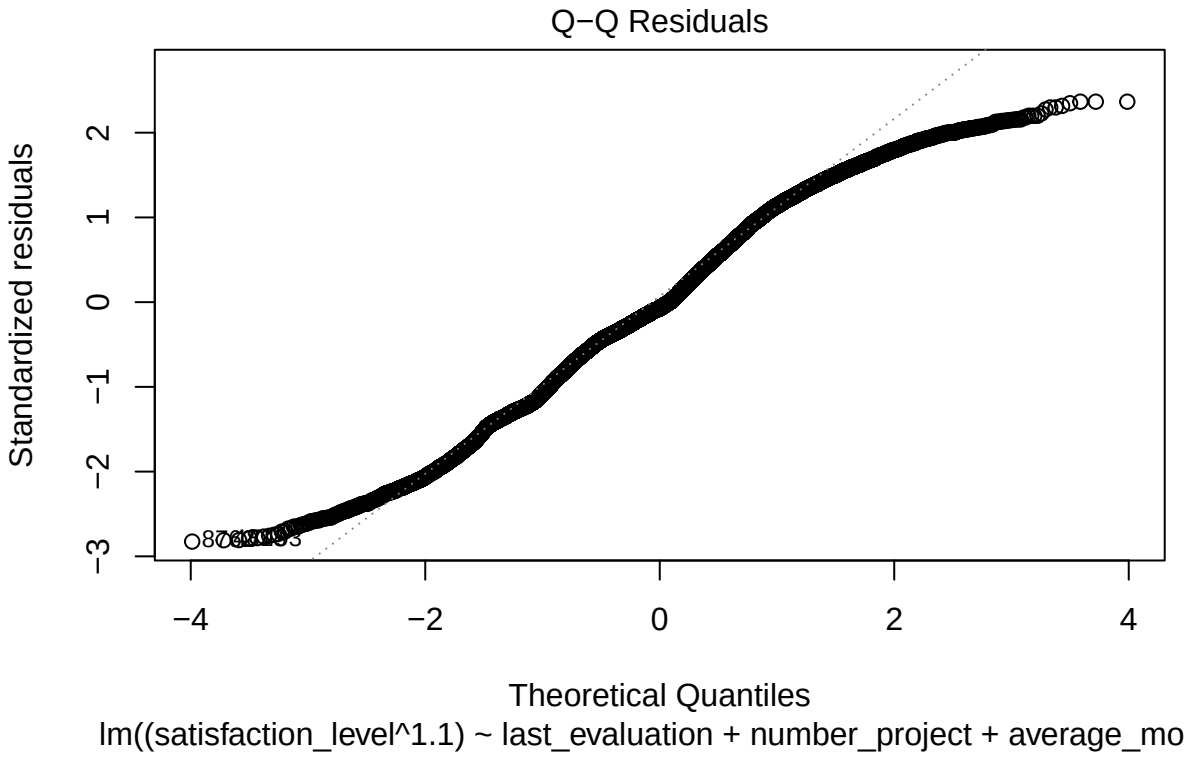



```
library(MASS)
lm1<-lm((satisfaction_level^1.1)~last_evaluation+number_project+average_montly_hours+time_spend_company+
boxcox(lm1)
lambda<-boxcox(lm1)
```


95%
log−Likelihood
−60000
−70000
−80000
−90000
−2
−1
0
1
2
λ

```
plot(lm1,1:2)
```


Residuals vs Fitted
Residuals
0.4
0.0
−0.4
−0.8
0.2
0.3
0.4
0.5
0.6
0.7
0.8
Fitted values
lm((satisfaction_level^1.1) ~ last_evaluation + number_project + average_mo ...




```
lm10<-lm(satisfaction_level~last_evaluation+number_project+average_montly_hours+time_spend_company+left+
summary(lm10)
```

```
##
## Call:
## lm(formula = satisfaction_level ~ last_evaluation + number_project +
##     average_montly_hours + time_spend_company + left + promotion_last_5years +
##     salary, data = data1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.64739 -0.13676 -0.01193  0.17003  0.52774
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             6.148e-01  1.156e-02  53.189  < 2e-16 ***
## last_evaluation         2.460e-01  1.167e-02  21.073  < 2e-16 ***
## number_project         -4.090e-02  1.691e-03 -24.184  < 2e-16 ***
## average_montly_hours    1.913e-04  4.126e-05   4.636 3.58e-06 ***
## time_spend_company     -5.525e-03  1.294e-03  -4.268 1.98e-05 ***
## left                   -2.241e-01  4.395e-03 -50.980  < 2e-16 ***
## promotion_last_5years   9.282e-03  1.272e-02   0.730   0.4654
## salarylow               1.200e-02  6.959e-03   1.724   0.0847 .
## salarymedium            1.306e-02  6.955e-03   1.878   0.0604 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2227 on 14990 degrees of freedom
## Multiple R-squared:  0.1982, Adjusted R-squared:  0.1977
## F-statistic:   463 on 8 and 14990 DF,  p-value: < 2.2e-16
```

```
table(data1$left)
```

```
##
##     0     1
## 11428  3571
```

```
library(MASS)
library(car)
```

```
## Loading required package: carData
```

```
summary(powerTransform(lm1))
```

```
## bcPower Transformation to Normality
##    Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd
## Y1     1.012           1       0.9833       1.0406
##
## Likelihood ratio test that transformation parameter is equal to 0
##  (log transformation)
##                             LRT df       pval
## LR test, lambda = (0) 6018.397  1 < 2.22e-16
##
## Likelihood ratio test that no transformation is needed
##                              LRT df    pval
## LR test, lambda = (1) 0.6756013  1 0.41111
```

```
library(car)
vif(lm1)
```

```
##      last_evaluation       number_project average_montly_hours
##             1.206278             1.313408             1.284216
##   time_spend_company                 left
##             1.067913             1.025957
```

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

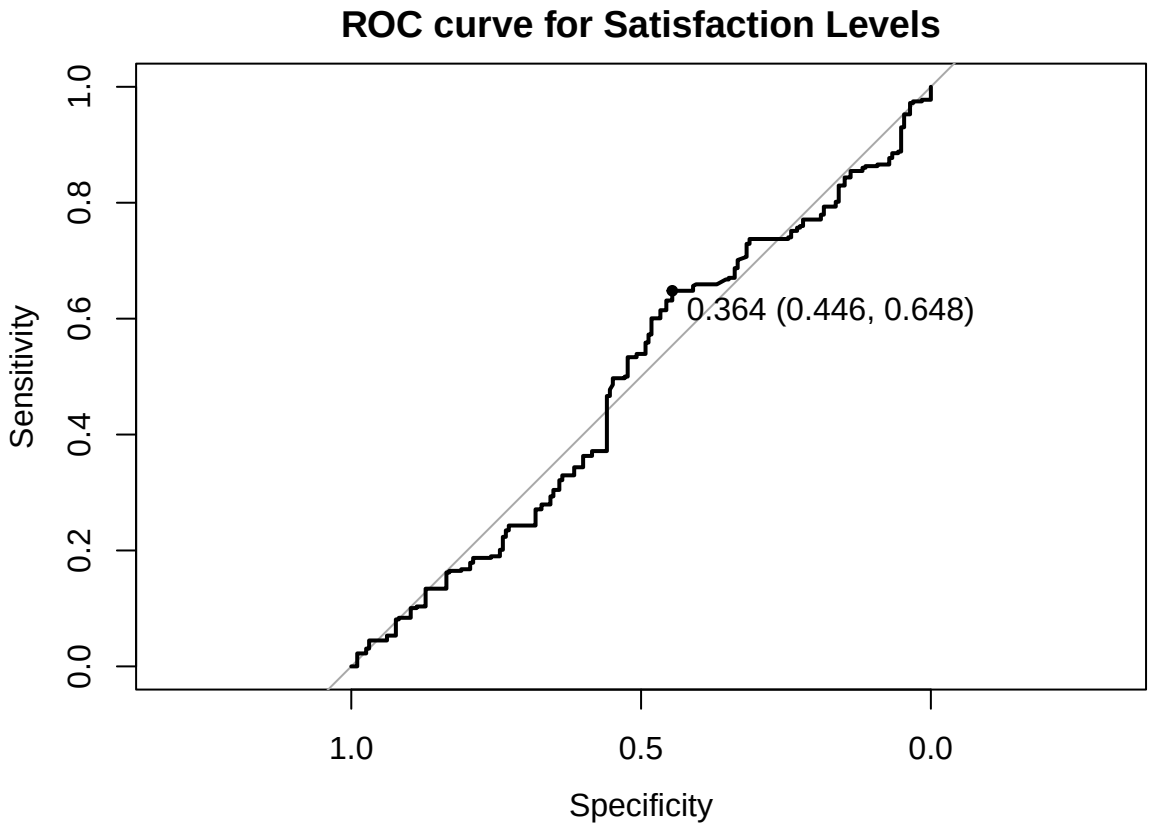
```
roc.curve <- roc(satisfaction_level ~ predict(lm1, type = "response"), data = data1)
```

```
## Warning in roc.default(response, predictors[, 1], ...): 'response' has more
## than two levels. Consider setting 'levels' explicitly or using 'multiclass.roc'
## instead
```

```
## Setting levels: control = 0.09, case = 0.1
```

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

```
plot(roc.curve, main = "ROC curve for Satisfaction Levels", print.thres = "best")
```




```
auc(roc.curve)
```

```
## Area under the curve: 0.4934
```

```
data1$Department<-as.numeric(as.factor(data1$Department))
data1$salary<-as.numeric(as.factor(data1$salary))
```

```
head(data1)
```

```
##   satisfaction_level last_evaluation number_project average_montly_hours
## 1               0.38            0.53              2                   157
## 2               0.80            0.86              5                   262
## 3               0.11            0.88              7                   272
## 4               0.72            0.87              5                   223
## 5               0.37            0.52              2                   159
## 6               0.41            0.50              2                   153
##   time_spend_company Work_accident left promotion_last_5years Department salary
## 1                  3             0    1                     0          8      2
## 2                  6             0    1                     0          8      3
## 3                  4             0    1                     0          8      3
## 4                  5             0    1                     0          8      2
## 5                  3             0    1                     0          8      2
## 6                  3             0    1                     0          8      2
```

```
mlr_model <- lm(satisfaction_level ~ ., data =data1)
summary(mlr_model)
```

```
##
## Call:
## lm(formula = satisfaction_level ~ ., data = data1)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.65320 -0.13779 -0.01234  0.17071  0.52592
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             6.058e-01  1.275e-02  47.526  < 2e-16 ***
## last_evaluation         2.462e-01  1.167e-02  21.101  < 2e-16 ***
## number_project         -4.097e-02  1.691e-03 -24.223  < 2e-16 ***
## average_montly_hours    1.908e-04  4.126e-05   4.623 3.80e-06 ***
## time_spend_company     -5.519e-03  1.292e-03  -4.270 1.96e-05 ***
## Work_accident          -9.185e-05  5.237e-03  -0.018   0.9860
## left                   -2.233e-01  4.388e-03 -50.886  < 2e-16 ***
## promotion_last_5years   8.945e-03  1.268e-02   0.705   0.4806
## Department              1.573e-03  6.631e-04   2.373   0.0177 *
## salary                  4.055e-03  2.906e-03   1.395   0.1630
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2227 on 14989 degrees of freedom
## Multiple R-squared:  0.1984, Adjusted R-squared:  0.1979
## F-statistic: 412.1 on 9 and 14989 DF,  p-value: < 2.2e-16
```

```
data$Department<-as.factor(data$Department)
data$salary<-as.factor(data$salary)
```

```
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
# split data into training and test sets
train_index <- sample(nrow(data1), nrow(data1) * 0.75)
train <- data1[train_index, ]
test <- data1[-train_index, ]
mlr_model <- lm(satisfaction_level ~ ., data = train)
predictions <- predict(mlr_model, newdata = test)
rmse <- RMSE(predictions, test$satisfaction_level)
rmse
```

```
## [1] 0.2212009
```

```
summary(mlr_model)
```

```
##
## Call:
## lm(formula = satisfaction_level ~ ., data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.64895 -0.13570 -0.01032  0.17253  0.52185
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             6.012e-01  1.478e-02  40.675  < 2e-16 ***
```

```
## last_evaluation        2.399e-01  1.353e-02  17.730  < 2e-16 ***
## number_project        -4.046e-02  1.974e-03 -20.497  < 2e-16 ***
## average_montly_hours   2.491e-04  4.777e-05   5.215 1.87e-07 ***
## time_spend_company    -5.931e-03  1.513e-03  -3.921 8.88e-05 ***
## Work_accident         -4.844e-03  6.067e-03  -0.798   0.4246
## left                  -2.208e-01  5.103e-03 -43.279  < 2e-16 ***
## promotion_last_5years  3.594e-03  1.433e-02   0.251   0.8020
## Department             1.475e-03  7.678e-04   1.921   0.0548 .
## salary                 3.950e-03  3.370e-03   1.172   0.2411
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2232 on 11239 degrees of freedom
## Multiple R-squared:  0.1919, Adjusted R-squared:  0.1912
## F-statistic: 296.5 on 9 and 11239 DF,  p-value: < 2.2e-16
```

```
library(caret)
# split data into training and test sets
train_index <- sample(nrow(data1), nrow(data1) * 0.75)
train <- data1[train_index, ]
test <- data1[-train_index, ]
knn_model <- train(satisfaction_level ~ ., data = train,
method = "knn", trControl = trainControl(method = "cv",
number = 5), tuneLength = 10)
predictions <- predict(knn_model, newdata = test)
rmse <- RMSE(predictions, test$satisfaction_level)
rmse
```

```
## [1] 0.1977994
```

```
knn_model
```

```
## k-Nearest Neighbors
##
## 11249 samples
##     9 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 8998, 9000, 9000, 9000, 8998
## Resampling results across tuning parameters:
##
##   k   RMSE       Rsquared   MAE
##    5  0.2028767  0.3552761  0.1496793
##    7  0.2009082  0.3586237  0.1496936
##    9  0.1997137  0.3609254  0.1503834
##   11  0.1992942  0.3607152  0.1508052
##   13  0.1992305  0.3595621  0.1514978
##   15  0.1992596  0.3583880  0.1519872
##   17  0.1997530  0.3547459  0.1528668
##   19  0.2003440  0.3506159  0.1536929
##   21  0.2011000  0.3456501  0.1546474
##   23  0.2017491  0.3412722  0.1553646
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was k = 13.
```

```
library(xgboost)
# split data into training and test sets
train_index <- sample(nrow(data1), nrow(data1) * 0.75)
train <- data1[train_index, ]
test <- data1[-train_index, ]
dtrain <- xgb.DMatrix(data = as.matrix(train[, -1]), label = train$satisfaction_level)
dtest <- xgb.DMatrix(data = as.matrix(test[, -1]), label = test$satisfaction_level)
```

```
library(randomForest)
```

```
## randomForest 4.7-1.1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
# split data into training and test sets
train_index <- sample(nrow(data1), nrow(data1) * 0.80)
train <- data1[train_index, ]
test <- data1[-train_index, ]
rf_model <- randomForest(satisfaction_level ~ .,
                         data = train, ntree = 20)
predicted_values <- predict(rf_model, newdata = test)
rmse <- sqrt(mean((test$satisfaction_level - predicted_values)^2))
rmse
```

```
## [1] 0.1717567
```

```
rf_model
```

```
##
## Call:
##  randomForest(formula = satisfaction_level ~ ., data = train,      ntree = 20)
##                Type of random forest: regression
##                      Number of trees: 20
## No. of variables tried at each split: 3
##
##           Mean of squared residuals: 0.03029174
##                     % Var explained: 50.73
```

```
data$Department<-as.numeric(as.factor(data$Department))
data$salary<-as.numeric(as.factor(data$salary))
```

```
library(xgboost)
# split data into training and test sets
train_index <- sample(nrow(data), nrow(data) * 0.75)
train <- data[train_index, ]
test <- data[-train_index, ]
xgb_model <- xgboost(data = as.matrix(train[, -which(names(train)
== "satisfaction_level")]), label = train$satisfaction_level,
nrounds = 500)
```

```
## [1]  train-rmse:0.231694
```

```
## [2]  train-rmse:0.205845
## [3]  train-rmse:0.190960
## [4]  train-rmse:0.182636
## [5]  train-rmse:0.178094
## [6]  train-rmse:0.175009
## [7]  train-rmse:0.173001
## [8]  train-rmse:0.170988
## [9]  train-rmse:0.169444
## [10] train-rmse:0.168323
## [11] train-rmse:0.166817
## [12] train-rmse:0.165660
## [13] train-rmse:0.164699
## [14] train-rmse:0.163806
## [15] train-rmse:0.163215
## [16] train-rmse:0.162786
## [17] train-rmse:0.161984
## [18] train-rmse:0.161772
## [19] train-rmse:0.160859
## [20] train-rmse:0.160173
## [21] train-rmse:0.159571
## [22] train-rmse:0.159164
## [23] train-rmse:0.158472
## [24] train-rmse:0.157759
## [25] train-rmse:0.157056
## [26] train-rmse:0.156046
## [27] train-rmse:0.155487
## [28] train-rmse:0.155122
## [29] train-rmse:0.154443
## [30] train-rmse:0.153791
## [31] train-rmse:0.153481
## [32] train-rmse:0.153382
## [33] train-rmse:0.152876
## [34] train-rmse:0.151969
## [35] train-rmse:0.151239
## [36] train-rmse:0.150977
## [37] train-rmse:0.149940
## [38] train-rmse:0.149350
## [39] train-rmse:0.148962
## [40] train-rmse:0.148160
## [41] train-rmse:0.147312
## [42] train-rmse:0.146709
## [43] train-rmse:0.146263
## [44] train-rmse:0.145493
## [45] train-rmse:0.145282
## [46] train-rmse:0.144997
## [47] train-rmse:0.144779
## [48] train-rmse:0.144092
## [49] train-rmse:0.143503
## [50] train-rmse:0.143294
## [51] train-rmse:0.142889
## [52] train-rmse:0.142333
## [53] train-rmse:0.141690
## [54] train-rmse:0.141336
## [55] train-rmse:0.140833
```

```
## [56] train-rmse:0.140207
## [57] train-rmse:0.139925
## [58] train-rmse:0.139720
## [59] train-rmse:0.138928
## [60] train-rmse:0.138696
## [61] train-rmse:0.138057
## [62] train-rmse:0.137664
## [63] train-rmse:0.137442
## [64] train-rmse:0.136889
## [65] train-rmse:0.136646
## [66] train-rmse:0.136382
## [67] train-rmse:0.135729
## [68] train-rmse:0.135216
## [69] train-rmse:0.134712
## [70] train-rmse:0.134516
## [71] train-rmse:0.134244
## [72] train-rmse:0.133663
## [73] train-rmse:0.133359
## [74] train-rmse:0.133253
## [75] train-rmse:0.132604
## [76] train-rmse:0.132436
## [77] train-rmse:0.132223
## [78] train-rmse:0.132060
## [79] train-rmse:0.131706
## [80] train-rmse:0.131649
## [81] train-rmse:0.131074
## [82] train-rmse:0.130757
## [83] train-rmse:0.130250
## [84] train-rmse:0.129885
## [85] train-rmse:0.129353
## [86] train-rmse:0.129271
## [87] train-rmse:0.128875
## [88] train-rmse:0.128604
## [89] train-rmse:0.128171
## [90] train-rmse:0.127851
## [91] train-rmse:0.127389
## [92] train-rmse:0.126878
## [93] train-rmse:0.126356
## [94] train-rmse:0.126134
## [95] train-rmse:0.125986
## [96] train-rmse:0.125815
## [97] train-rmse:0.125492
## [98] train-rmse:0.125048
## [99] train-rmse:0.124637
## [100]    train-rmse:0.124167
## [101]    train-rmse:0.123618
## [102]    train-rmse:0.123535
## [103]    train-rmse:0.123180
## [104]    train-rmse:0.122885
## [105]    train-rmse:0.122501
## [106]    train-rmse:0.122201
## [107]    train-rmse:0.121879
## [108]    train-rmse:0.121265
## [109]    train-rmse:0.120903
```

```
## [110]    train-rmse:0.120752
## [111]    train-rmse:0.120472
## [112]    train-rmse:0.120350
## [113]    train-rmse:0.119804
## [114]    train-rmse:0.119525
## [115]    train-rmse:0.119158
## [116]    train-rmse:0.118723
## [117]    train-rmse:0.118642
## [118]    train-rmse:0.118218
## [119]    train-rmse:0.117933
## [120]    train-rmse:0.117629
## [121]    train-rmse:0.117287
## [122]    train-rmse:0.117099
## [123]    train-rmse:0.116890
## [124]    train-rmse:0.116511
## [125]    train-rmse:0.116316
## [126]    train-rmse:0.116133
## [127]    train-rmse:0.115990
## [128]    train-rmse:0.115625
## [129]    train-rmse:0.115589
## [130]    train-rmse:0.115518
## [131]    train-rmse:0.115354
## [132]    train-rmse:0.115126
## [133]    train-rmse:0.114746
## [134]    train-rmse:0.114527
## [135]    train-rmse:0.114187
## [136]    train-rmse:0.113902
## [137]    train-rmse:0.113773
## [138]    train-rmse:0.113585
## [139]    train-rmse:0.113374
## [140]    train-rmse:0.113158
## [141]    train-rmse:0.112987
## [142]    train-rmse:0.112606
## [143]    train-rmse:0.112442
## [144]    train-rmse:0.112124
## [145]    train-rmse:0.111881
## [146]    train-rmse:0.111755
## [147]    train-rmse:0.111592
## [148]    train-rmse:0.111175
## [149]    train-rmse:0.110991
## [150]    train-rmse:0.110695
## [151]    train-rmse:0.110359
## [152]    train-rmse:0.110174
## [153]    train-rmse:0.110143
## [154]    train-rmse:0.109907
## [155]    train-rmse:0.109855
## [156]    train-rmse:0.109761
## [157]    train-rmse:0.109397
## [158]    train-rmse:0.109346
## [159]    train-rmse:0.109294
## [160]    train-rmse:0.109231
## [161]    train-rmse:0.108958
## [162]    train-rmse:0.108856
## [163]    train-rmse:0.108832
```

```
## [164]    train-rmse:0.108373
## [165]    train-rmse:0.108232
## [166]    train-rmse:0.108159
## [167]    train-rmse:0.108057
## [168]    train-rmse:0.107760
## [169]    train-rmse:0.107475
## [170]    train-rmse:0.107168
## [171]    train-rmse:0.106860
## [172]    train-rmse:0.106604
## [173]    train-rmse:0.106268
## [174]    train-rmse:0.106109
## [175]    train-rmse:0.105917
## [176]    train-rmse:0.105563
## [177]    train-rmse:0.105497
## [178]    train-rmse:0.105459
## [179]    train-rmse:0.105371
## [180]    train-rmse:0.105201
## [181]    train-rmse:0.105101
## [182]    train-rmse:0.104857
## [183]    train-rmse:0.104623
## [184]    train-rmse:0.104338
## [185]    train-rmse:0.103985
## [186]    train-rmse:0.103850
## [187]    train-rmse:0.103632
## [188]    train-rmse:0.103390
## [189]    train-rmse:0.103122
## [190]    train-rmse:0.102713
## [191]    train-rmse:0.102349
## [192]    train-rmse:0.101950
## [193]    train-rmse:0.101609
## [194]    train-rmse:0.101309
## [195]    train-rmse:0.101021
## [196]    train-rmse:0.100887
## [197]    train-rmse:0.100685
## [198]    train-rmse:0.100469
## [199]    train-rmse:0.100427
## [200]    train-rmse:0.100277
## [201]    train-rmse:0.100255
## [202]    train-rmse:0.100212
## [203]    train-rmse:0.099842
## [204]    train-rmse:0.099555
## [205]    train-rmse:0.099361
## [206]    train-rmse:0.099234
## [207]    train-rmse:0.098945
## [208]    train-rmse:0.098581
## [209]    train-rmse:0.098479
## [210]    train-rmse:0.098420
## [211]    train-rmse:0.098106
## [212]    train-rmse:0.097863
## [213]    train-rmse:0.097759
## [214]    train-rmse:0.097640
## [215]    train-rmse:0.097433
## [216]    train-rmse:0.097389
## [217]    train-rmse:0.097300
```

```
## [218]    train-rmse:0.097253
## [219]    train-rmse:0.097032
## [220]    train-rmse:0.096715
## [221]    train-rmse:0.096460
## [222]    train-rmse:0.096370
## [223]    train-rmse:0.096155
## [224]    train-rmse:0.095966
## [225]    train-rmse:0.095670
## [226]    train-rmse:0.095399
## [227]    train-rmse:0.095163
## [228]    train-rmse:0.094947
## [229]    train-rmse:0.094724
## [230]    train-rmse:0.094446
## [231]    train-rmse:0.094286
## [232]    train-rmse:0.094091
## [233]    train-rmse:0.093939
## [234]    train-rmse:0.093555
## [235]    train-rmse:0.093255
## [236]    train-rmse:0.093200
## [237]    train-rmse:0.092958
## [238]    train-rmse:0.092858
## [239]    train-rmse:0.092720
## [240]    train-rmse:0.092325
## [241]    train-rmse:0.092278
## [242]    train-rmse:0.092110
## [243]    train-rmse:0.092087
## [244]    train-rmse:0.091935
## [245]    train-rmse:0.091827
## [246]    train-rmse:0.091797
## [247]    train-rmse:0.091753
## [248]    train-rmse:0.091547
## [249]    train-rmse:0.091419
## [250]    train-rmse:0.091098
## [251]    train-rmse:0.090820
## [252]    train-rmse:0.090611
## [253]    train-rmse:0.090450
## [254]    train-rmse:0.090331
## [255]    train-rmse:0.090032
## [256]    train-rmse:0.089860
## [257]    train-rmse:0.089748
## [258]    train-rmse:0.089536
## [259]    train-rmse:0.089273
## [260]    train-rmse:0.089191
## [261]    train-rmse:0.089114
## [262]    train-rmse:0.088882
## [263]    train-rmse:0.088762
## [264]    train-rmse:0.088567
## [265]    train-rmse:0.088330
## [266]    train-rmse:0.088231
## [267]    train-rmse:0.088063
## [268]    train-rmse:0.087923
## [269]    train-rmse:0.087850
## [270]    train-rmse:0.087596
## [271]    train-rmse:0.087288
```

```
## [272]    train-rmse:0.087080
## [273]    train-rmse:0.086936
## [274]    train-rmse:0.086864
## [275]    train-rmse:0.086537
## [276]    train-rmse:0.086342
## [277]    train-rmse:0.086144
## [278]    train-rmse:0.086004
## [279]    train-rmse:0.085978
## [280]    train-rmse:0.085696
## [281]    train-rmse:0.085527
## [282]    train-rmse:0.085500
## [283]    train-rmse:0.085222
## [284]    train-rmse:0.084965
## [285]    train-rmse:0.084766
## [286]    train-rmse:0.084649
## [287]    train-rmse:0.084505
## [288]    train-rmse:0.084366
## [289]    train-rmse:0.084229
## [290]    train-rmse:0.084084
## [291]    train-rmse:0.083869
## [292]    train-rmse:0.083834
## [293]    train-rmse:0.083559
## [294]    train-rmse:0.083523
## [295]    train-rmse:0.083332
## [296]    train-rmse:0.083095
## [297]    train-rmse:0.082946
## [298]    train-rmse:0.082912
## [299]    train-rmse:0.082878
## [300]    train-rmse:0.082853
## [301]    train-rmse:0.082648
## [302]    train-rmse:0.082416
## [303]    train-rmse:0.082176
## [304]    train-rmse:0.082062
## [305]    train-rmse:0.081781
## [306]    train-rmse:0.081516
## [307]    train-rmse:0.081307
## [308]    train-rmse:0.081092
## [309]    train-rmse:0.080857
## [310]    train-rmse:0.080666
## [311]    train-rmse:0.080445
## [312]    train-rmse:0.080344
## [313]    train-rmse:0.080150
## [314]    train-rmse:0.079866
## [315]    train-rmse:0.079781
## [316]    train-rmse:0.079605
## [317]    train-rmse:0.079378
## [318]    train-rmse:0.079153
## [319]    train-rmse:0.078980
## [320]    train-rmse:0.078774
## [321]    train-rmse:0.078717
## [322]    train-rmse:0.078628
## [323]    train-rmse:0.078388
## [324]    train-rmse:0.078191
## [325]    train-rmse:0.077955
```

```
## [326]    train-rmse:0.077722
## [327]    train-rmse:0.077525
## [328]    train-rmse:0.077425
## [329]    train-rmse:0.077269
## [330]    train-rmse:0.077145
## [331]    train-rmse:0.076983
## [332]    train-rmse:0.076900
## [333]    train-rmse:0.076753
## [334]    train-rmse:0.076679
## [335]    train-rmse:0.076504
## [336]    train-rmse:0.076342
## [337]    train-rmse:0.076261
## [338]    train-rmse:0.076242
## [339]    train-rmse:0.076091
## [340]    train-rmse:0.076053
## [341]    train-rmse:0.075885
## [342]    train-rmse:0.075721
## [343]    train-rmse:0.075569
## [344]    train-rmse:0.075453
## [345]    train-rmse:0.075404
## [346]    train-rmse:0.075151
## [347]    train-rmse:0.075044
## [348]    train-rmse:0.074943
## [349]    train-rmse:0.074712
## [350]    train-rmse:0.074552
## [351]    train-rmse:0.074364
## [352]    train-rmse:0.074292
## [353]    train-rmse:0.074157
## [354]    train-rmse:0.074125
## [355]    train-rmse:0.073995
## [356]    train-rmse:0.073866
## [357]    train-rmse:0.073807
## [358]    train-rmse:0.073624
## [359]    train-rmse:0.073565
## [360]    train-rmse:0.073514
## [361]    train-rmse:0.073474
## [362]    train-rmse:0.073321
## [363]    train-rmse:0.073107
## [364]    train-rmse:0.072924
## [365]    train-rmse:0.072855
## [366]    train-rmse:0.072728
## [367]    train-rmse:0.072641
## [368]    train-rmse:0.072544
## [369]    train-rmse:0.072413
## [370]    train-rmse:0.072330
## [371]    train-rmse:0.072301
## [372]    train-rmse:0.072113
## [373]    train-rmse:0.072010
## [374]    train-rmse:0.071926
## [375]    train-rmse:0.071899
## [376]    train-rmse:0.071701
## [377]    train-rmse:0.071524
## [378]    train-rmse:0.071491
## [379]    train-rmse:0.071303
```

```
## [380]    train-rmse:0.071130
## [381]    train-rmse:0.070976
## [382]    train-rmse:0.070806
## [383]    train-rmse:0.070687
## [384]    train-rmse:0.070536
## [385]    train-rmse:0.070447
## [386]    train-rmse:0.070330
## [387]    train-rmse:0.070120
## [388]    train-rmse:0.069972
## [389]    train-rmse:0.069951
## [390]    train-rmse:0.069785
## [391]    train-rmse:0.069773
## [392]    train-rmse:0.069744
## [393]    train-rmse:0.069626
## [394]    train-rmse:0.069565
## [395]    train-rmse:0.069344
## [396]    train-rmse:0.069213
## [397]    train-rmse:0.069028
## [398]    train-rmse:0.068886
## [399]    train-rmse:0.068756
## [400]    train-rmse:0.068613
## [401]    train-rmse:0.068465
## [402]    train-rmse:0.068417
## [403]    train-rmse:0.068276
## [404]    train-rmse:0.068180
## [405]    train-rmse:0.068056
## [406]    train-rmse:0.067894
## [407]    train-rmse:0.067840
## [408]    train-rmse:0.067628
## [409]    train-rmse:0.067615
## [410]    train-rmse:0.067541
## [411]    train-rmse:0.067460
## [412]    train-rmse:0.067312
## [413]    train-rmse:0.067129
## [414]    train-rmse:0.067020
## [415]    train-rmse:0.067006
## [416]    train-rmse:0.066957
## [417]    train-rmse:0.066808
## [418]    train-rmse:0.066755
## [419]    train-rmse:0.066701
## [420]    train-rmse:0.066637
## [421]    train-rmse:0.066533
## [422]    train-rmse:0.066339
## [423]    train-rmse:0.066198
## [424]    train-rmse:0.066112
## [425]    train-rmse:0.066002
## [426]    train-rmse:0.065932
## [427]    train-rmse:0.065812
## [428]    train-rmse:0.065670
## [429]    train-rmse:0.065631
## [430]    train-rmse:0.065581
## [431]    train-rmse:0.065556
## [432]    train-rmse:0.065429
## [433]    train-rmse:0.065271
```

```
## [434]    train-rmse:0.065037
## [435]    train-rmse:0.065009
## [436]    train-rmse:0.064943
## [437]    train-rmse:0.064905
## [438]    train-rmse:0.064734
## [439]    train-rmse:0.064547
## [440]    train-rmse:0.064492
## [441]    train-rmse:0.064398
## [442]    train-rmse:0.064244
## [443]    train-rmse:0.064031
## [444]    train-rmse:0.063984
## [445]    train-rmse:0.063948
## [446]    train-rmse:0.063926
## [447]    train-rmse:0.063825
## [448]    train-rmse:0.063780
## [449]    train-rmse:0.063768
## [450]    train-rmse:0.063578
## [451]    train-rmse:0.063411
## [452]    train-rmse:0.063334
## [453]    train-rmse:0.063280
## [454]    train-rmse:0.063170
## [455]    train-rmse:0.063045
## [456]    train-rmse:0.062944
## [457]    train-rmse:0.062861
## [458]    train-rmse:0.062733
## [459]    train-rmse:0.062639
## [460]    train-rmse:0.062481
## [461]    train-rmse:0.062431
## [462]    train-rmse:0.062263
## [463]    train-rmse:0.062107
## [464]    train-rmse:0.062055
## [465]    train-rmse:0.061966
## [466]    train-rmse:0.061917
## [467]    train-rmse:0.061781
## [468]    train-rmse:0.061764
## [469]    train-rmse:0.061754
## [470]    train-rmse:0.061644
## [471]    train-rmse:0.061590
## [472]    train-rmse:0.061464
## [473]    train-rmse:0.061305
## [474]    train-rmse:0.061167
## [475]    train-rmse:0.061091
## [476]    train-rmse:0.061024
## [477]    train-rmse:0.060967
## [478]    train-rmse:0.060879
## [479]    train-rmse:0.060790
## [480]    train-rmse:0.060691
## [481]    train-rmse:0.060614
## [482]    train-rmse:0.060462
## [483]    train-rmse:0.060342
## [484]    train-rmse:0.060280
## [485]    train-rmse:0.060222
## [486]    train-rmse:0.060145
## [487]    train-rmse:0.060030
```

```
## [488]    train-rmse:0.060001
## [489]    train-rmse:0.059960
## [490]    train-rmse:0.059900
## [491]    train-rmse:0.059870
## [492]    train-rmse:0.059733
## [493]    train-rmse:0.059681
## [494]    train-rmse:0.059543
## [495]    train-rmse:0.059454
## [496]    train-rmse:0.059370
## [497]    train-rmse:0.059321
## [498]    train-rmse:0.059294
## [499]    train-rmse:0.059288
## [500]    train-rmse:0.059210
```

```
predicted_values <- predict(xgb_model,newdata =
as.matrix(test[, -which(names(test) == "satisfaction_level")]))
rmse <- sqrt(mean((test$satisfaction_level - predicted_values)^2))
rmse
```

```
## [1] 0.1896665
```