DAT-4253 LM 4.1 - KNN

AUTHOR
Aaron Younger

```r
## R type
suppressWarnings(RNGversion("3.5.3"))
```

```r
##Libraries
library(readxl)
library(dplyr)
library(tidyverse)
library(DataExplorer)
library(caret)
library(pROC)
library(gains)
library(flextable)
library(SmartEDA)
library(dlookr)
library(ggplot2)
```

```r
library(readxl)
HR_Data <- read_excel("jaggia_ba_2e_ch12_data.xlsx", sheet = "HR_Data")
View(HR_Data)

HR_Score <- read_excel("jaggia_ba_2e_ch12_data.xlsx", sheet = "HR_Score")
View(HR_Score)
```

Business Understanding

Daniel Lara, a human resources manager at a large tech consulting firm, has been reading about using analytics to predict the success of new employees. With the fast-changing nature of the tech industry, some employees have had difficulties staying current in their field and have missed the opportunity to be promoted into a management position. Daniel is particularly interested in whether or not a new employee is likely to be promoted into a management role after 10 years with the company. In the accompanying data file, he gathers information on 300 current employees who have worked for the firm for at least 10 years. The information was based on the job application that the employees provided when they originally applied for a job at the firm. For each employee, the following variables are listed: Promoted (1 if promoted within 10 years, 0 otherwise), GPA (college GPA at graduation), Sports (number of athletic activities during college), and Leadership (number of leadership roles in student organizations).

Business Goal

- The Goal of this model is to use appropriate predictors to predict if a new employee will be promoted in 10 years.

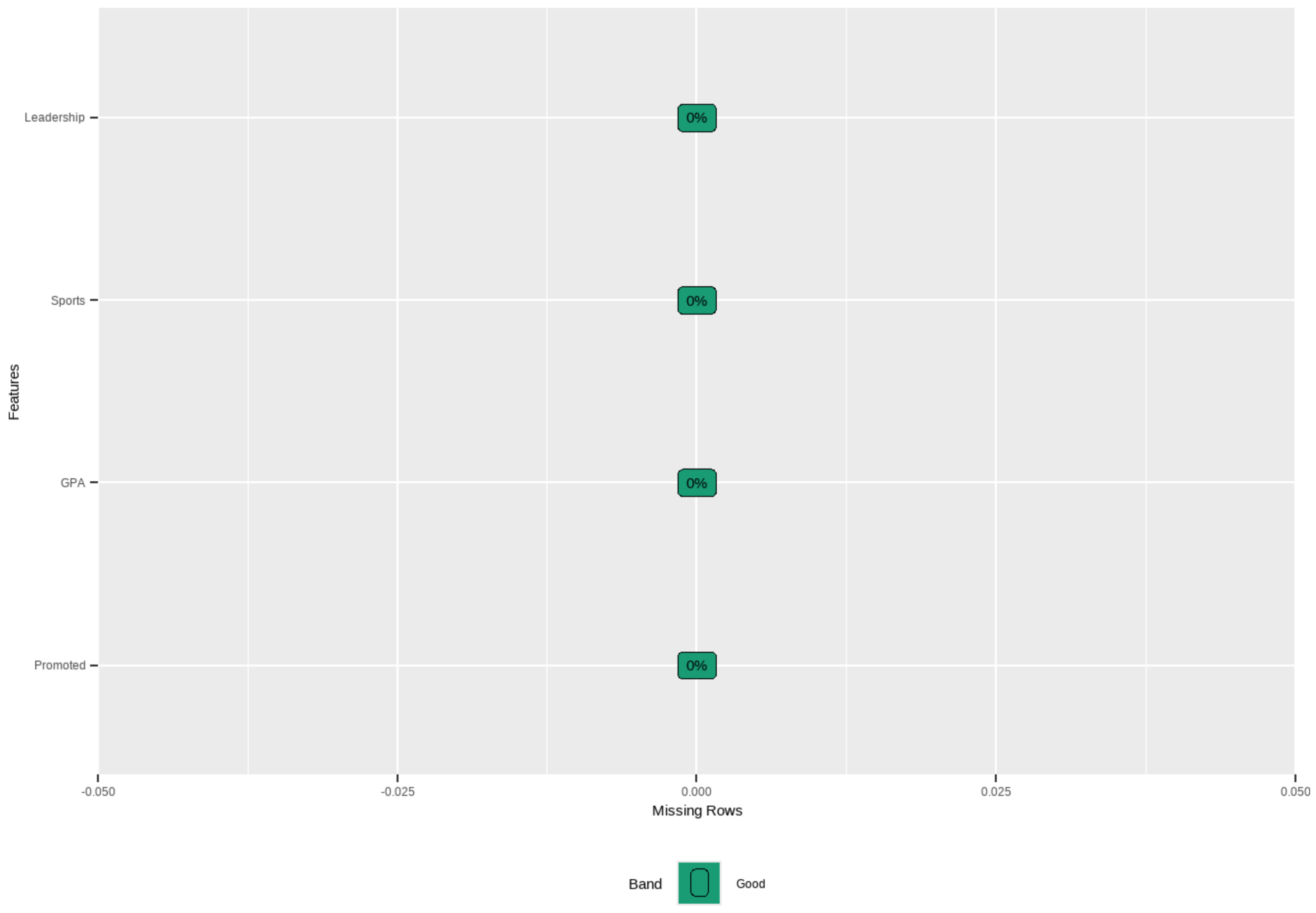
Data Understanding

EDA

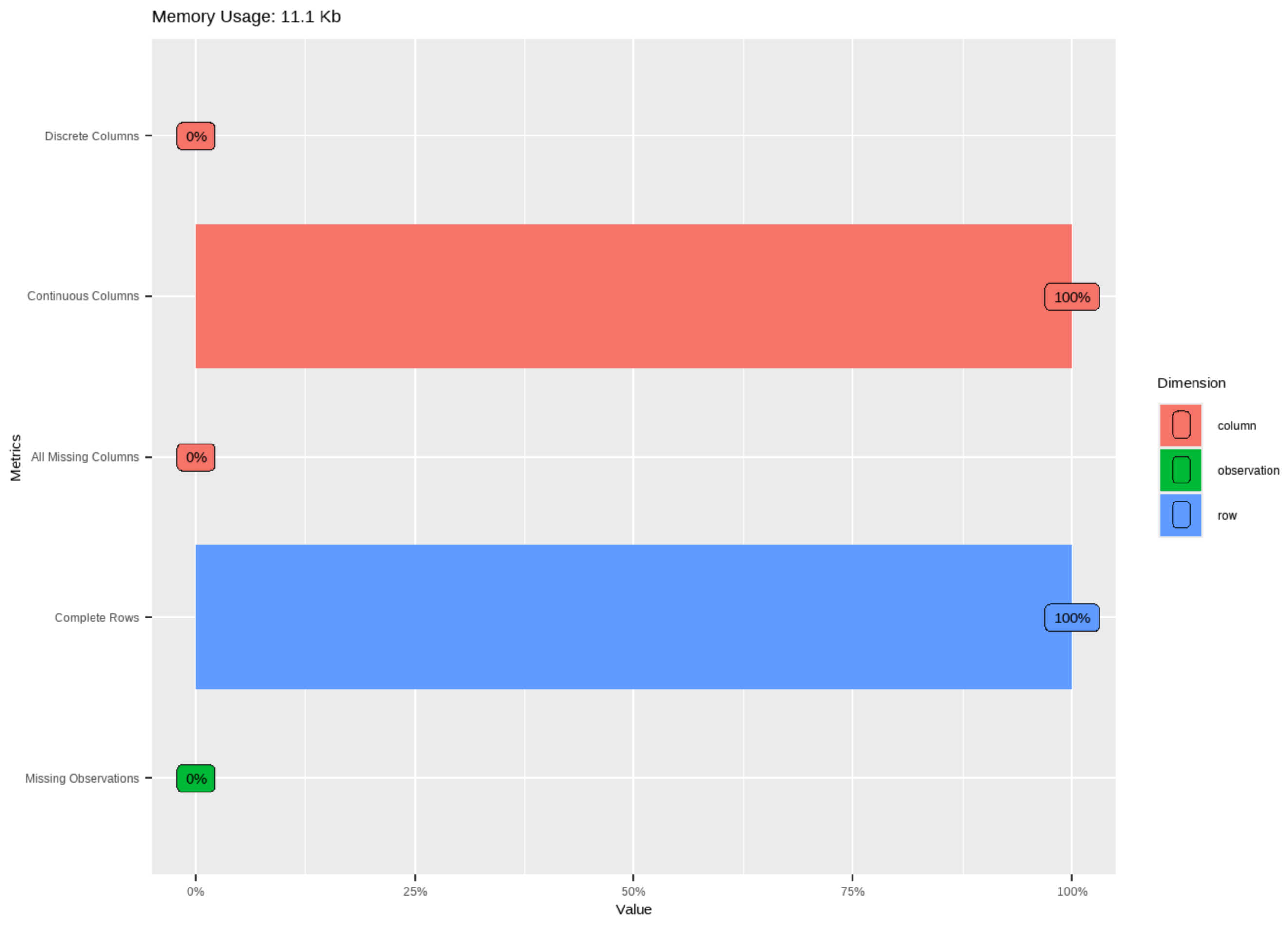
```
HR_Data %>% str()
```

```
tibble [300 × 4] (S3: tbl_df/tbl/data.frame)
 $ Promoted  : num [1:300] 0 1 0 1 0 1 1 1 1 1 ...
 $ GPA       : num [1:300] 3.28 3.93 3.17 3.87 3.2 3.34 3.01 3.87 2.52 3.14 ...
 $ Sports    : num [1:300] 0 6 5 1 5 5 3 4 2 4 ...
 $ Leadership: num [1:300] 2 3 1 4 0 0 1 2 2 2 ...
```

```
HR_Data %>% plot_missing()
```



```
HR_Data %>% plot_intro()
```



```
HR_Data %>% head()
```

```
# A tibble: 6 × 4
  Promoted   GPA Sports Leadership
     <dbl> <dbl>  <dbl>      <dbl>
1        0  3.28      0          2
2        1  3.93      6          3
3        0  3.17      5          1
4        1  3.87      1          4
5        0  3.2       5          0
6        1  3.34      5          0
```

```
HR_Data %>% tail()
```

```
# A tibble: 6 × 4
  Promoted   GPA Sports Leadership
     <dbl> <dbl>  <dbl>      <dbl>
1        1  3.78      4          2
2        0  2.69      2          1
3        0  3.36      0          1
4        0  3.93      4          4
```

```
5         0  3.57      4           2
6         0  3.54      5           0
```

```
HR_Data %>%
  group_by(Promoted) %>%
  count() %>%
  ungroup() %>%
  mutate(Percent = n / sum(n) *100)
```

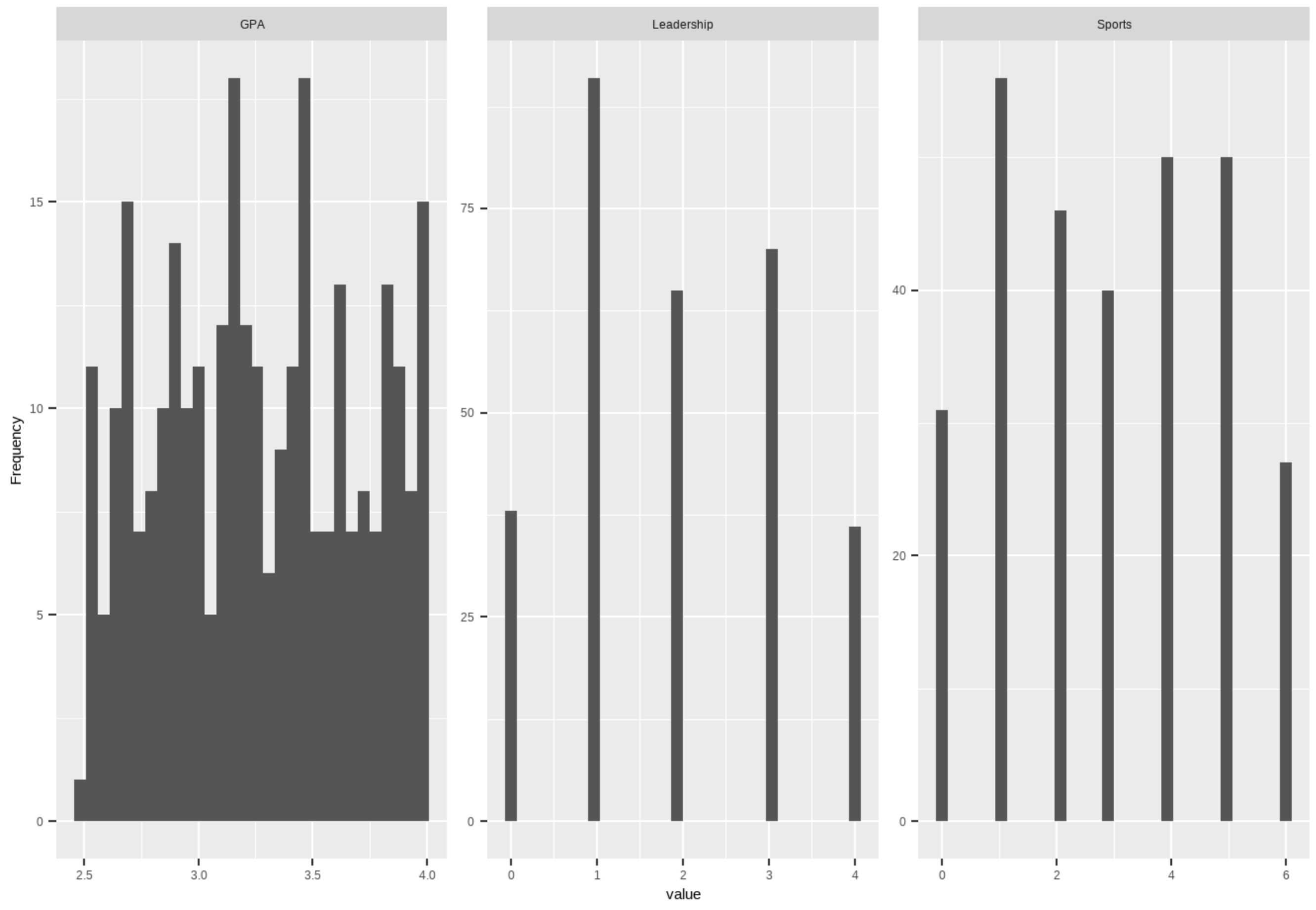
```
# A tibble: 2 × 3
  Promoted     n Percent
     <dbl> <int>   <dbl>
1        0    92    30.7
2        1   208    69.3
```

EDA Comments:

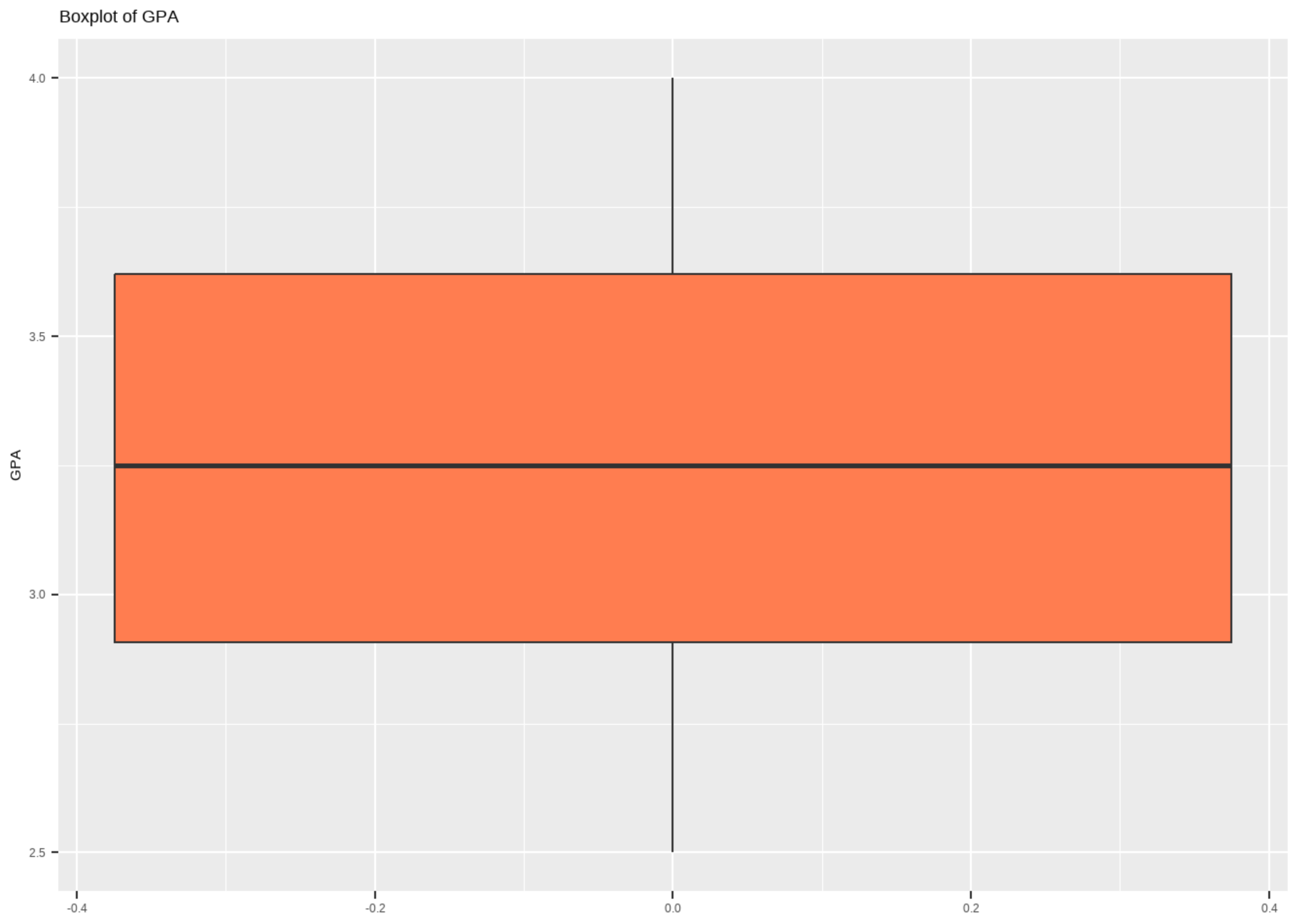
This Dataset is imbalanced between the classes of the Promoted Variable. The promoted variable is 70% made up of the 1 class and 30% made up of the 0 class. This means the model may favor the majority class (1) because it reduces error overall, specificity will be something to note when evaluating the model.

Variable EDA

```
HR_Data %>% plot_histogram()
```



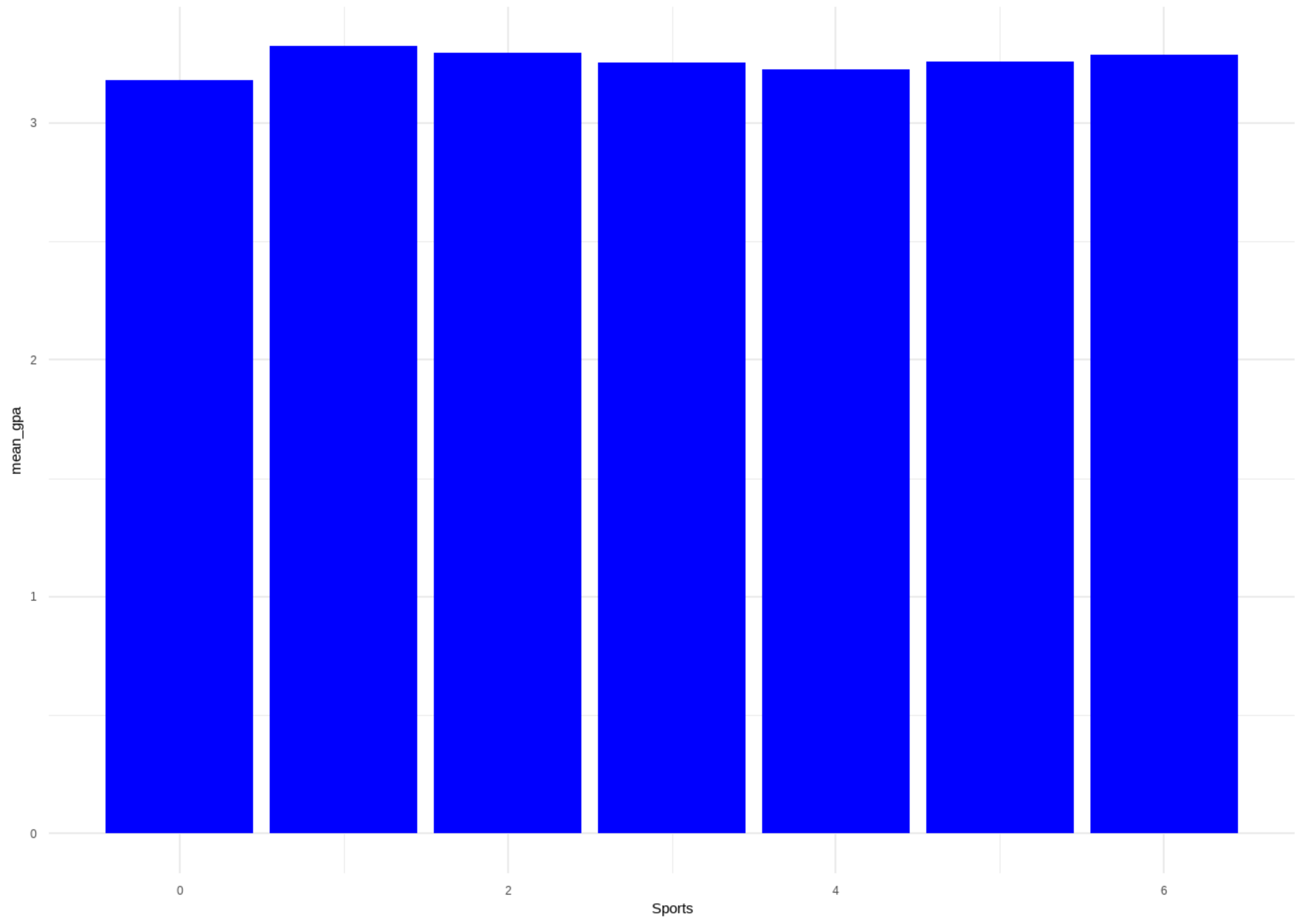
```
## No outliers in GPA variable or other variables
ggplot(HR_Data, aes(y=GPA)) +
  geom_boxplot(fill = "coral")+
  labs(title = "Boxplot of GPA",
       y = "GPA")
```



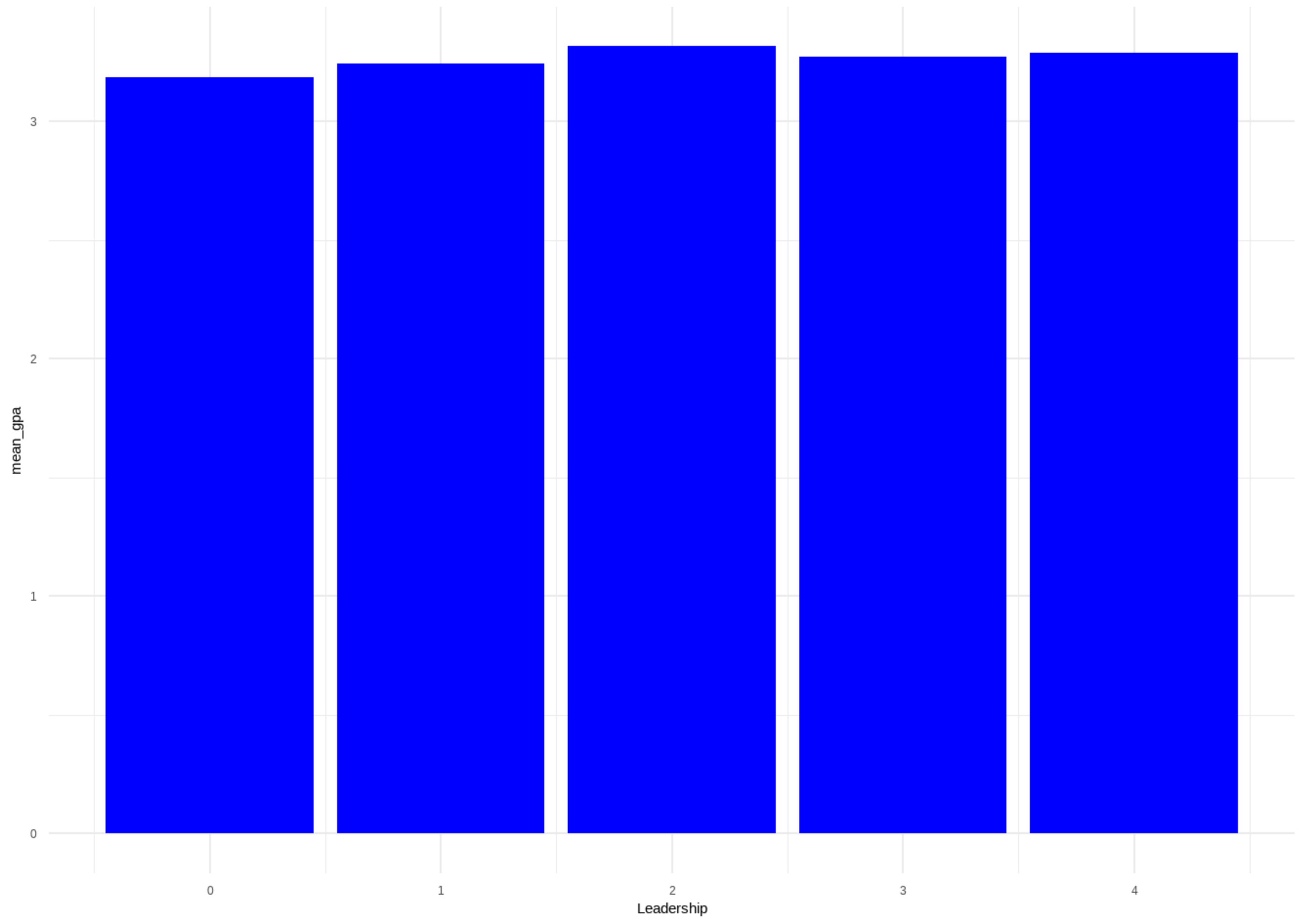
```r
diagnose_outlier(HR_Data) %>% flextable()
```

variables	outliers_cnt	outliers_ratio	outliers_mean	with_mean	without_mean
Promoted	0	0		0.6933333	0.6933333
GPA	0	0		3.2651667	3.2651667
Sports	0	0		2.9333333	2.9333333
Leadership	0	0		1.9166667	1.9166667

```r
## GPA remains relatively constant over Sports variable
HR_Data %>%
  group_by(Sports) %>%
  summarise(mean_gpa = mean(GPA)) %>%
  select(Sports, mean_gpa) %>%
  ggplot(aes(x=Sports, y=mean_gpa))+
  geom_bar(stat = "identity", fill = "blue") +
  theme_minimal()
```



```r
## GPA remains relatively constant over Leadership varaible
HR_Data %>%
  group_by(Leadership) %>%
  summarise(mean_gpa = mean(GPA)) %>%
  select(Leadership, mean_gpa) %>%
  ggplot(aes(x=Leadership, y=mean_gpa))+
  geom_bar(stat = "identity", fill = "blue") +
  theme_minimal()
```



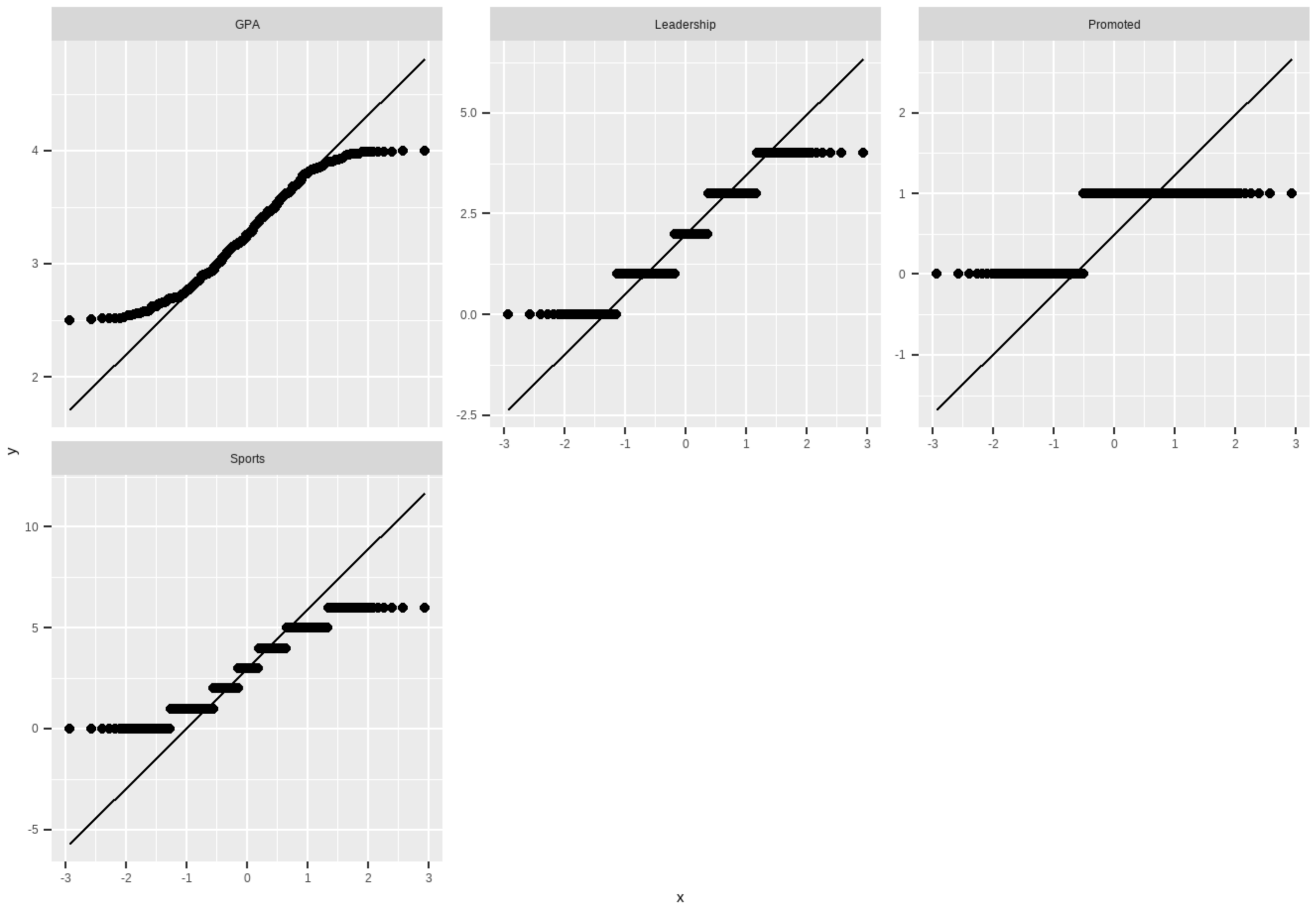
```
## Looking at the mean across both classes of promoted, the average GPA is not different across
         classes but a higher average sports and leadership is in class one of promoted.
HR_Data %>%
  group_by(Promoted) %>%
  summarise(mean_gpa = mean(GPA),
            mean_sports = mean(Sports),
            mean_Leadership = mean(Leadership))
```

```
# A tibble: 2 × 4
  Promoted mean_gpa mean_sports mean_Leadership
     <dbl>    <dbl>       <dbl>           <dbl>
1        0     3.28        1.99            1.25
2        1     3.26        3.35            2.21
```

Variable EDA Comments:

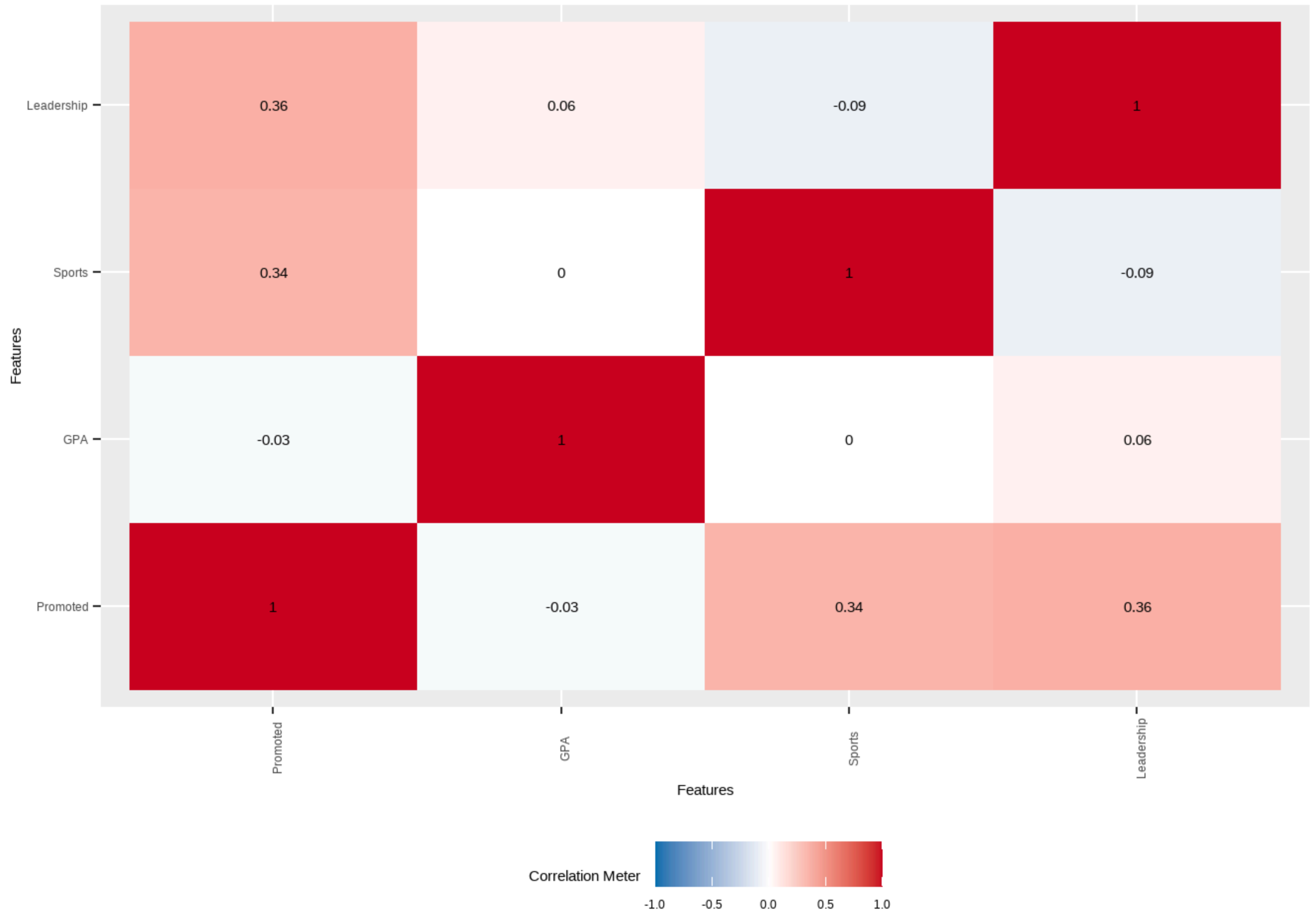
This dataset contains no outliers. When looking at relationships among variables to the dependent variable we find that employees promoted tend to have a higher amount of sports activities in college and leadership roles in student organizations. This is something to keep in mind when evaluating model scoring.

```
HR_Data %>% plot_qq()
```



Since KNN is a non parametric model the data does not need to be distributed a certain way.

```r
HR_Data %>% plot_correlation()
```



Similar to insights found above, sports and leadership have a positive correlation to being promoted.

Data Preperation

```
## Create Standardized Data Frame
HR_Data1 <- scale(HR_Data[2:4])
HR_Data1 <- data.frame(HR_Data1, HR_Data$Promoted)
colnames(HR_Data1)[4] <- 'Promoted'
view(HR_Data1)

## Make Promoted a factor
HR_Data1$Promoted <- as.factor(HR_Data$Promoted)
HR_Data1 %>% str()
```

```
'data.frame':	300 obs. of  4 variables:
 $ GPA       : num  0.0343 1.5383 -0.2202 1.3995 -0.1508 ...
 $ Sports    : num  -1.57 1.64 1.11 -1.04 1.11 ...
 $ Leadership: num  0.0676 0.8783 -0.7432 1.689 -1.5539 ...
 $ Promoted  : Factor w/ 2 levels "0","1": 1 2 1 2 1 2 2 2 2 2 ...
```

Data Preparation Comments

Before KNN Modeling the data needs to be standardized to ensure that different units do not interfere with the distance based calculations. The dependent variable is also turned into a factor so that the KNN model will correctly do classification.

Modeling

```
## Partition Dataset
set.seed(1)
myindex <- createDataPartition(HR_Data1$Promoted, p=0.6, list = FALSE)
trainset <- HR_Data1[myindex,]
testset <- HR_Data1[-myindex,]
```

Before running the KNN model the data is partitioned into a training dataset and a test dataset. The training dataset will be used to fit the model. The test set is used to evaluate to see how well the model generalizes.

```
##
myCtrl <- trainControl(method = "CV", number = 10)

myGrid <- expand.grid(.k=c(2:10))

set.seed(1)
KNN_fit <- train(Promoted ~ ., data = trainset, method = "knn", trControl=myCtrl, tuneGrid =
         myGrid)
KNN_fit
```

```
k-Nearest Neighbors

181 samples
  3 predictor
  2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 163, 163, 163, 162, 162, 163, ...
Resampling results across tuning parameters:

  k   Accuracy   Kappa
   2  0.7592879  0.4241839
   3  0.7201066  0.3049729
   4  0.7475576  0.3748290
   5  0.7361541  0.3450899
   6  0.7305986  0.3100753
   7  0.7809598  0.4422326
   8  0.7695562  0.4095484
   9  0.7470072  0.3515362
  10  0.7581183  0.3750668
```

```
Accuracy was used to select the optimal model using the largest value.
The final value used for the model was k = 7.
```

Modeling Comments

The Model was set up using 10-fold cross validation and hyper parameter tuning. 10-fold cross validation is used to prevent overfitting and hyper parameter tuning systematically chose neighbors varying from the closest 2 through 10. Each number of neighbors was evaluated during the cross validation and used to show what number of neighbors gave the best results. The KNN Model was then run using the train dataset. Based on this process, the optimal model was found at k=7. This tuned KNN model, trained on the full training dataset, was selected for further evaluation on the test dataset.

Evaluation

```r
## Evaluate Model
KNN_predict <- predict(KNN_fit, newdata = testset)
confusionMatrix(KNN_predict, testset$Promoted, positive = '1')
```

```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 22 15
         1 14 68

               Accuracy : 0.7563
                 95% CI : (0.6691, 0.8303)
    No Information Rate : 0.6975
    P-Value [Acc > NIR] : 0.09551

                  Kappa : 0.427

 Mcnemar's Test P-Value : 1.00000

            Sensitivity : 0.8193
            Specificity : 0.6111
         Pos Pred Value : 0.8293
         Neg Pred Value : 0.5946
             Prevalence : 0.6975
         Detection Rate : 0.5714
   Detection Prevalence : 0.6891
      Balanced Accuracy : 0.7152

       'Positive' Class : 1
```

```r
KNN_Class_prob <- predict(KNN_fit, newdata = testset, type ='prob')
tail(KNN_Class_prob)
```

```
            0         1
114 0.0000000 1.0000000
115 0.0000000 1.0000000
116 0.7142857 0.2857143
117 0.8571429 0.1428571
118 1.0000000 0.0000000
119 0.0000000 1.0000000
```

Evaluation Comments: Confusion Matrix

The Model was then tested on the test dataset. A confusion matrix was made to help show the results of how well the model performed. The model showed an accuracy of 75.6% which is the proportion of all correct predictions. The 95% CI shows that with 95% confidence the accuracy will lie within the range of (0.6691, 0.8303). The No Information Rate shows the model does better than trivial guessing as the accuracy 0.7563 is over the NIR value of 0.6975. The P-Value shows the model is not statistically significant 0.09551 > 0.05. The sensitivity rate of 0.8193 shows the model is good at identifying promoted employees. The specificity rate of 0.6111 shows the model struggles to identify non-promoted employees. This can be due to the imbalanced dataset towards 0 classes (non-promoted employees). The Positive predictive value shows of those predicted as promoted, 83% were correct which shows good precision. However the negative predictive value shows of those predicted as not promoted, only 59% were correct showing weaker precision towards the 0 class. A prevalence of 0.6975 shows the proportion of 1 classes in the dataset, so approximately 70%. The Detection rate of 0.571 showed the proportion of all records that were correctly predicted as positives. Detection Prevalence, 0.689, shows the records that were predicted as positive, this is close to the actual prevalence amount. And finally the Balanced accuracy of 0.715 shows the model is reasonably balanced but stronger at predicting class 1.

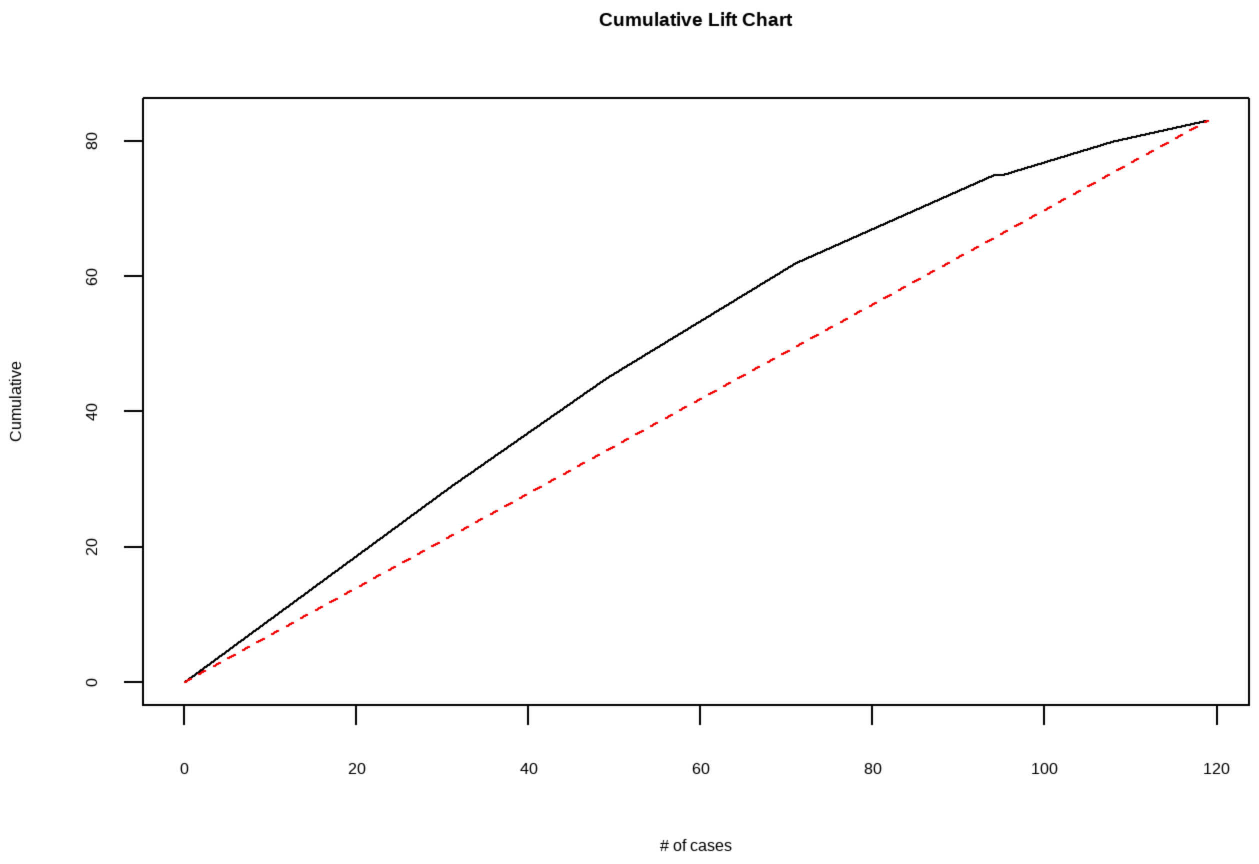
```r
testset$Promoted <- as.numeric(as.character(testset$Promoted))
head(testset$Promoted)
```

```
[1] 1 1 1 1 1 0
```

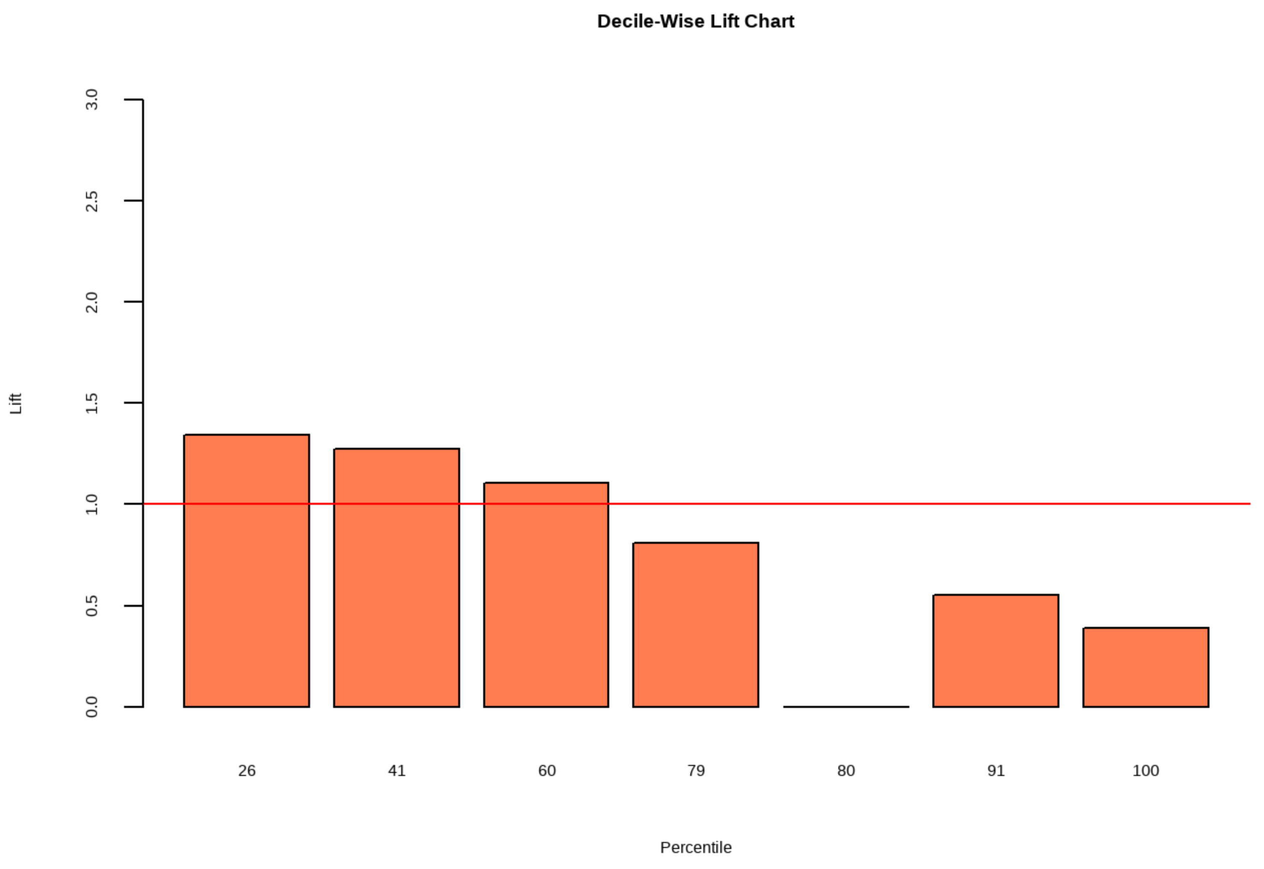
```r
## Gains table
gains_table <- gains(testset$Promoted, KNN_Class_prob[,2])
gains_table
```

```
Depth                            Cume   Cume Pct                     Mean
 of              Cume     Mean   Mean   of Total    Lift   Cume     Model
File     N         N      Resp   Resp       Resp   Index   Lift     Score
-------------------------------------------------------------------------
  26    31        31      0.94   0.94      34.9%     134    134      1.00
  41    18        49      0.89   0.92      54.2%     127    132      0.86
  60    22        71      0.77   0.87      74.7%     111    125      0.71
  79    23     94      0.57      0.80      90.4%     81    114      0.49
  80     1     95      0.00      0.79      90.4%      0    113      0.38
  91    13    108      0.38      0.74      96.4%     55    106      0.28
 100    11    119      0.27      0.70     100.0%     39    100      0.10
  NA    NA     NA        NA        NA        NA%     NA     NA        NA
  NA    NA     NA        NA        NA        NA%     NA     NA        NA
  NA    NA     NA        NA        NA        NA%     NA     NA        NA
```

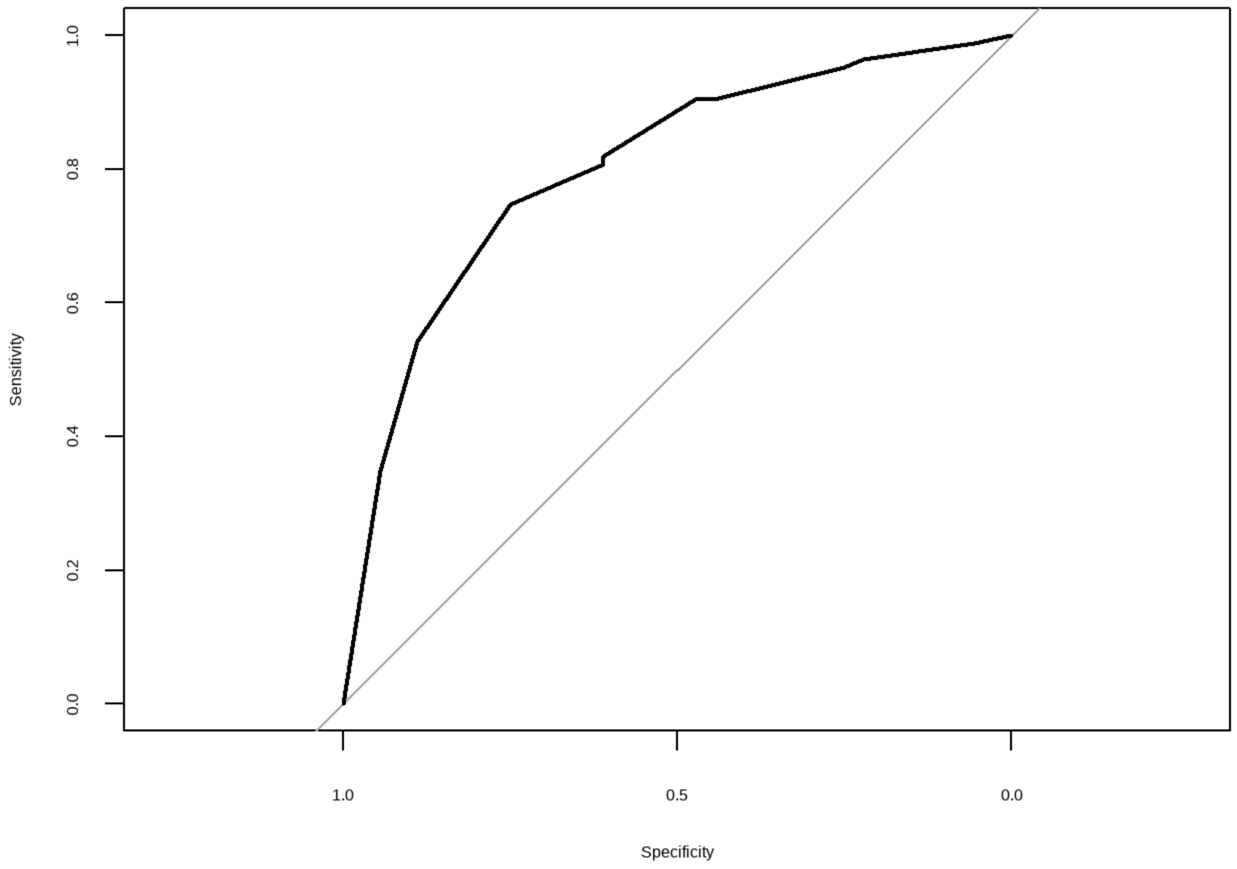
```r
## Cumulative Lift Chart
plot(c(0, gains_table$cume.pct.of.total*sum(testset$Promoted))~c(0, gains_table$cume.obs), xlab
         = "# of cases", ylab = "Cumulative", main="Cumulative Lift Chart", type="l")
lines(c(0, sum(testset$Promoted))~c(0, dim(testset)[1]), col="red", lty=2)
```



```r
## Decile wise lift chart
barplot(gains_table$mean.resp/mean(testset$Promoted), names.arg=gains_table$depth,
         xlab="Percentile", ylab="Lift", ylim=c(0,3), col = "coral", main="Decile-Wise Lift
         Chart")
abline(h=c(1),col="red")
```



```
## ROC Chart with AUC
roc_object <- roc(testset$Promoted, KNN_Class_prob[,2])
plot.roc(roc_object)
```



```
auc(roc_object)
```

```
Area under the curve: 0.8015
```

Evaluation Comments: Charts and Graphs

The Model was further evaluated based of charts and graphs. Both Cumulative and Decile-wise lift charts show that the model captures larger positive cases in upper segments of the data. The Cumulative lift chart also shows that the model performs better than random chance. The Roc curve indicates that the model has good discriminative ability between promoted and non-promoted employees. The Auc value of 0.8015 supports what the Roc curve shows. Finally, the gains table shows that the top 40% of the dataset captures more than half the promoted employees. This makes the model valuable for prioritization who to market to.

Deployment

```
## Score the Model
preprocess <- preProcess(HR_Data[ , 2:4], method = c("center", "scale"))

HR_Score1 <- predict(preprocess, HR_Score)
```

```
KNN_Score <- predict(KNN_fit, newdata = HR_Score1)

HR_Score <- data.frame(HR_Score, KNN_Score)
print(HR_Score)
```

```
   GPA Sports Leadership KNN_Score
1 2.75      1          2         0
2 2.92      0          4         1
3 3.01      4          2         1
4 4.00      5          3         1
5 3.91      1          0         0
6 3.20      2          1         1
```

```
table(HR_Score$GPA, HR_Score$KNN_Score)
```

```
      
       0 1
  2.75 1 0
  2.92 0 1
  3.01 0 1
  3.2  0 1
  3.91 1 0
  4    0 1
```

```
table(HR_Score$Sports, HR_Score$KNN_Score)
```

```
   
    0 1
  0 0 1
  1 2 0
  2 0 1
  4 0 1
  5 0 1
```

```
table(HR_Score$Leadership, HR_Score$KNN_Score)
```

```
   
    0 1
  0 1 0
  1 0 1
  2 1 1
  3 0 1
  4 0 1
```

```
HR_Score %>%
  group_by(KNN_Score) %>%
  summarise(meanGPA = mean(GPA),
            meanSports = mean(Sports),
            meanleadership = mean(Leadership))
```

```
# A tibble: 2 × 4
  KNN_Score meanGPA meanSports meanleadership
  <fct>       <dbl>      <dbl>          <dbl>
1 0            3.33       1               1
2 1            3.28       2.75            2.5
```

Comments on Deployment

Lastly, the scoring dataset was used to generate promotion predictions. Since in the scoring dataset promotion is unknown, the dataset is not used to assess accuracy but highlight what the predictor variables look like to the predicted promotion types. The scoring dataset shows that people with a higher level of sports and leadership involvement are predicted to be promoted over those without. It also shows that GPA is not a big indicator of whether someone gets promoted or not.

Final Model Recommendations

This KNN model demonstrates moderate predictive ability especially when predicting employees that will be promoted, though it struggles with identifying non-promotions. This model highlights the importance of extracurricular involvement relative to promotion success. Future models should focus on addressing class imbalance and validating the model on more datasets.