Individual Assignment 3: Classification using K-Nearest Neighbours and Naive Bayes by Shimony Agrawal

Download the necessary packages for classification using KNN and Naive Bayes.

```
# install.packages("DBI")
# install.packages("odbc")
# install.packages("dplyr")
# install.packages("tidyverse")
# install.packages("lubridate")
# install.packages("ggplot2")
# install.packages("ISLR")
# install.packages("caret")
# install.packages("forecast")
# install.packages("corrplot")
# install.packages ("visualize")
# install.packages("FNN")
# install.packages("e1071")

library(DBI)
```

```
## Warning: package 'DBI' was built under R version 4.0.2
```

```
library(odbc)
```

```
## Warning: package 'odbc' was built under R version 4.0.2
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.0.2

## -- Attaching packages --------------------------------------------------------------- tidyverse 1.3.0

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.1     v dplyr   1.0.0
## v tidyr   1.1.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## Warning: package 'ggplot2' was built under R version 4.0.2

## Warning: package 'dplyr' was built under R version 4.0.2

## -- Conflicts ------------------------------------------------------------------ tidyverse_conflicts()
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 4.0.2
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```r
library(dplyr)
library(ggplot2)
library(ISLR)
```

```
## Warning: package 'ISLR' was built under R version 4.0.2
```

```r
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.0.2
```

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

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```r
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.0.2
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```r
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.0.2
```

```
## corrplot 0.84 loaded
```

```r
library(visualize)
```

```
## Warning: package 'visualize' was built under R version 4.0.2
```

```
library(FNN)
```

```
## Warning: package 'FNN' was built under R version 4.0.2
```

```
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.0.2
```

```
Part 1: K-Nearest Neighbours
```

Task 1: Read the file.

```
employees = read.csv("/Users/shimonyagrawal/Desktop/Grad /Summer 2/AD699_Data Mining/RStudio/Assignment
employees <- employees[,c(2, 1, 3:12)]
```

Data type of variables

```
str(employees)
```

```
## 'data.frame':    1470 obs. of  12 variables:
##  $ Attrition              : chr  "Yes" "No" "Yes" "No" ...
##  $ Age                    : int  41 49 37 33 27 32 59 30 38 36 ...
##  $ DistanceFromHome       : int  1 8 2 3 2 2 3 24 23 27 ...
##  $ MonthlyIncome          : int  5993 5130 2090 2909 3468 3068 2670 2693 9526 5237 ...
##  $ NumCompaniesWorked     : int  8 1 6 1 9 0 4 1 0 6 ...
##  $ PercentSalaryHike      : int  11 23 15 11 12 13 20 22 21 13 ...
##  $ TotalWorkingYears      : int  8 10 7 8 6 8 12 1 10 17 ...
##  $ TrainingTimesLastYear  : int  0 3 3 3 3 2 3 2 2 3 ...
##  $ YearsAtCompany         : int  6 10 0 8 2 7 1 1 9 7 ...
##  $ YearsInCurrentRole     : int  4 7 0 7 2 7 0 0 7 7 ...
##  $ YearsSinceLastPromotion: int  0 1 0 3 2 3 0 0 1 7 ...
##  $ YearsWithCurrManager   : int  5 7 0 0 2 6 0 0 8 7 ...
```

```
employees$Attrition <- as.factor(employees$Attrition)

str(employees)
```

```
## 'data.frame':    1470 obs. of  12 variables:
##  $ Attrition              : Factor w/ 2 levels "No","Yes": 2 1 2 1 1 1 1 1 1 1 ...
##  $ Age                    : int  41 49 37 33 27 32 59 30 38 36 ...
##  $ DistanceFromHome       : int  1 8 2 3 2 2 3 24 23 27 ...
##  $ MonthlyIncome          : int  5993 5130 2090 2909 3468 3068 2670 2693 9526 5237 ...
##  $ NumCompaniesWorked     : int  8 1 6 1 9 0 4 1 0 6 ...
##  $ PercentSalaryHike      : int  11 23 15 11 12 13 20 22 21 13 ...
##  $ TotalWorkingYears      : int  8 10 7 8 6 8 12 1 10 17 ...
##  $ TrainingTimesLastYear  : int  0 3 3 3 3 2 3 2 2 3 ...
##  $ YearsAtCompany         : int  6 10 0 8 2 7 1 1 9 7 ...
##  $ YearsInCurrentRole     : int  4 7 0 7 2 7 0 0 7 7 ...
##  $ YearsSinceLastPromotion: int  0 1 0 3 2 3 0 0 1 7 ...
##  $ YearsWithCurrManager   : int  5 7 0 0 2 6 0 0 8 7 ...
```

Task 3: Are there any NAs in this dataset? Show the code that you used to find this out. If there are any NA values in any particular column, replace them with the median value for that column.

```
anyNA(employees$Attrition) # No NA values in the dataset.
```

```
## [1] FALSE
```

Comments: The dataset doesn't have NA values.

Task 4: Filter the original employees dataframe to create two new temporary dataframes. One of these dataframes should contain the records for employees who left the company, and the other dataframe should contain the records for employees who did not leave the company.Call the summary() function on each of these two dataframes that you just made.

Q: Based on the summary identify differences that you noticed between the summary stats for the employees who left and the stats for the employees who did not.For each of the major differences that you found, include a sentence or two of speculation about what why/how these factors might impact employees' decisions to stay or leave.

```
employees_left <- employees %>%
  filter (Attrition == 'Yes')

employees_stayed <- employees %>%
  filter (Attrition == 'No')

summary(employees_left)
```

```
##  Attrition      Age        DistanceFromHome MonthlyIncome   NumCompaniesWorked
##  No :  0    Min.   :18.00   Min.   : 1.00    Min.   : 1009   Min.   :0.000     
##  Yes:237    1st Qu.:28.00   1st Qu.: 3.00    1st Qu.: 2373   1st Qu.:1.000     
##             Median :32.00   Median : 9.00    Median : 3202   Median :1.000     
##             Mean   :33.61   Mean   :10.63    Mean   : 4787   Mean   :2.941     
##             3rd Qu.:39.00   3rd Qu.:17.00    3rd Qu.: 5916   3rd Qu.:5.000     
##             Max.   :58.00   Max.   :29.00    Max.   :19859   Max.   :9.000     
##  PercentSalaryHike TotalWorkingYears TrainingTimesLastYear YearsAtCompany
##  Min.   :11.0      Min.   : 0.000    Min.   :0.000         Min.   : 0.000  
##  1st Qu.:12.0      1st Qu.: 3.000    1st Qu.:2.000         1st Qu.: 1.000  
##  Median :14.0      Median : 7.000    Median :2.000         Median : 3.000  
##  Mean   :15.1      Mean   : 8.245    Mean   :2.624         Mean   : 5.131  
##  3rd Qu.:17.0      3rd Qu.:10.000    3rd Qu.:3.000         3rd Qu.: 7.000  
##  Max.   :25.0      Max.   :40.000    Max.   :6.000         Max.   :40.000  
##  YearsInCurrentRole YearsSinceLastPromotion YearsWithCurrManager
##  Min.   : 0.000     Min.   : 0.000          Min.   : 0.000      
##  1st Qu.: 0.000     1st Qu.: 0.000          1st Qu.: 0.000      
##  Median : 2.000     Median : 1.000          Median : 2.000      
##  Mean   : 2.903     Mean   : 1.945          Mean   : 2.852      
##  3rd Qu.: 4.000     3rd Qu.: 2.000          3rd Qu.: 5.000      
##  Max.   :15.000     Max.   :15.000          Max.   :14.000      
```

```
summary(employees_stayed)
```

```
##  Attrition       Age        DistanceFromHome MonthlyIncome   NumCompaniesWorked
##  No :1233   Min.   :18.00   Min.   : 1.000   Min.   : 1051   Min.   :0.000     
##  Yes:   0   1st Qu.:31.00   1st Qu.: 2.000   1st Qu.: 3211   1st Qu.:1.000
##             Median :36.00   Median : 7.000   Median : 5204   Median :2.000
##             Mean   :37.56   Mean   : 8.916   Mean   : 6833   Mean   :2.646
##             3rd Qu.:43.00   3rd Qu.:13.000   3rd Qu.: 8834   3rd Qu.:4.000
##             Max.   :60.00   Max.   :29.000   Max.   :19999   Max.   :9.000
##  PercentSalaryHike TotalWorkingYears TrainingTimesLastYear YearsAtCompany
##  Min.   :11.00     Min.   : 0.00     Min.   :0.000         Min.   : 0.000
##  1st Qu.:12.00     1st Qu.: 6.00     1st Qu.:2.000         1st Qu.: 3.000
##  Median :14.00     Median :10.00     Median :3.000         Median : 6.000
##  Mean   :15.23     Mean   :11.86     Mean   :2.833         Mean   : 7.369
##  3rd Qu.:18.00     3rd Qu.:16.00     3rd Qu.:3.000         3rd Qu.:10.000
##  Max.   :25.00     Max.   :38.00     Max.   :6.000         Max.   :37.000
##  YearsInCurrentRole YearsSinceLastPromotion YearsWithCurrManager
##  Min.   : 0.000     Min.   : 0.000          Min.   : 0.000
##  1st Qu.: 2.000     1st Qu.: 0.000          1st Qu.: 2.000
##  Median : 3.000     Median : 1.000          Median : 3.000
##  Mean   : 4.484     Mean   : 2.234          Mean   : 4.367
##  3rd Qu.: 7.000     3rd Qu.: 3.000          3rd Qu.: 7.000
##  Max.   :18.000     Max.   :15.000          Max.   :17.000
```

Comments: Out of the data of 1470 employees, 1233 stayed and 237 left. Based on the summaries, I found that age, distance from home, number of companies worked, training and promotion aren't the key indicators of employees leaving or staying in the company. I was a bit surprised to find similar statistics for promotion in the data given its importance in a professional career. On the other hand, monthly income played an important role. Employees who left the company earned on an average $2000 less than the employees who stayed. Clearly, if employees feel they aren't paid what they deserve; they will tend to have lower motivation to work. Moreover, employees who stayed have on an average worked more i.e 11 years as compared to those left - 8 years. Here, the experience factor comes into play where the more experienced employees tend to earn more and hence have a higher monthly income. Another key observation was that employees who left stayed in their current role for an average of 3 years as compared to those who stayed. A possible interpretation could be their dissatisfaction with their current roles and found better opportunities elsewhere. Lastly, employees who left were assigned to their current manager for an average of 3 years whereas those who stayed were assigned for 4 years. Workplace environment plays a great role in a person's decision to work for the company. These employees were certainly unhappy with their current managers and wanted to leave. Having a motivated and dedicated person in a leadership role highly influences the employee's decision to stay or leave.

Concluding, firstly, monthly income and years with manager are key influencers in the decision to work at the company or not as well as professional experience in the industry. Job satisfaction is also a key decison-maker for employees today. Companies working towards making a more flexible environment for their employees and making them feel valued are definitely succeeding. However, given the data of 1,470 employees only 16% employees left - which is still a small number and can be improved too.

Task 5: Using your assigned seed value (from Assignment 2), partition your entire dataset into training (60%) and validation (40%) sets.

```
nrow(employees)
```

```
## [1] 1470
```

```
set.seed(10) # Assigned Seed Value
employees_sample <- sample_n(employees, 1470)
Train_KNN <- slice(employees_sample, 1:882)
Valid_KNN <- slice(employees_sample, 883:1470)
```

Task 6a: Emma

Task 6b:Use the runif() function to give your person values for each of the numeric predictor attributes. Use the min and max values from your training set as the lower and upper boundaries for runif().

```
Emma_runif <- data.frame(Age = runif(1, min(Train_KNN$Age), max(Train_KNN$Age)),
                   DistanceFromHome = runif(1, min(Train_KNN$DistanceFromHome), max(Train_KNN$DistanceF
                   MonthlyIncome = runif(1, min(Train_KNN$MonthlyIncome), max(Train_KNN$MonthlyIncome))
                   NumCompaniesWorked = runif(1, min(Train_KNN$NumCompaniesWorked), max(Train_KNN$NumCom
                   PercentSalaryHike = runif(1, min(Train_KNN$PercentSalaryHike), max(Train_KNN$Percent
                   TotalWorkingYears = runif(1, min(Train_KNN$TotalWorkingYears), max(Train_KNN$TotalWo
                   TrainingTimesLastYear = runif(1, min(Train_KNN$TrainingTimesLastYear), max(Train_KNN
                   YearsAtCompany = runif(1, min(Train_KNN$YearsAtCompany), max(Train_KNN$YearsAtCompan
                   YearsInCurrentRole = runif(1, min(Train_KNN$YearsInCurrentRole), max(Train_KNN$Years
                   YearsSinceLastPromotion = runif(1, min(Train_KNN$YearsSinceLastPromotion), max(Train_
                   YearsWithCurrManager = runif(1, min(Train_KNN$YearsWithCurrManager), max(Train_KNN$Ye

Emma <- data.frame(Age = 50.94084,
                   DistanceFromHome = 5.155811,
                   MonthlyIncome = 5233.086,
                   NumCompaniesWorked = 3.535845,
                   PercentSalaryHike = 24.89131,
                   TotalWorkingYears = 26.91134,
                   TrainingTimesLastYear = 3.311238,
                   YearsAtCompany = 38.72965,
                   YearsInCurrentRole = 7.547142,
                   YearsSinceLastPromotion = 3.1872,
                   YearsWithCurrManager = 12.10349)
```

Task 7: Normalize your data using the preProcess() function from the caret package

```
# Normalizing the dataset

Train.norm.df <- Train_KNN
Valid.norm.df <- Valid_KNN
employees.norm.df <- employees


norm.values <- preProcess(Train_KNN[, 2:12], method=c("center", "scale"))

Train.norm.df[, 2:12] <- predict(norm.values, Train_KNN[, 2:12])
Valid.norm.df[, 2:12] <- predict(norm.values, Valid_KNN[, 2:12])
employees.norm.df[, 2:12] <- predict(norm.values, employees[, 2:12])

new.norm.df <- predict(norm.values, Emma)
```

Task 8: Using the knn() function from the FNN package, and using a k-value of 7, generate a predicted classification for your employee.

Q: What outcome category was he or she predicted to belong to? Also, who were your person's 7 nearest neighbors? How many of them left the company, and how many stayed?

```
KNN <- knn(train = Train.norm.df[, 2:12], test = new.norm.df,
          cl = Train.norm.df[, 1], k = 7)
```

```
KNN
```

```
## [1] No
## attr(,"nn.index")
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7]
## [1,]  541  429   30  415  665  292  180
## attr(,"nn.dist")
##          [,1]     [,2]     [,3]     [,4]    [,5]     [,6]     [,7]
## [1,] 3.909966 4.316404 4.511436 4.570719 4.57336 4.758622 4.766593
## Levels: No
```

```
neighbours <- Train_KNN[c(541, 429, 30, 665, 292, 180),]
neighbours
```

```
##     Attrition Age DistanceFromHome MonthlyIncome NumCompaniesWorked
## 541        No  45                7          5210                  1
## 429        No  53                2         15427                  2
## 30         No  55               26         19586                  1
## 665        No  44                7         10248                  3
## 292        No  43                6          4081                  1
## 180        No  49                6         13966                  2
##     PercentSalaryHike TotalWorkingYears TrainingTimesLastYear YearsAtCompany
## 541                18                24                     2             24
## 429                16                31                     3             25
## 30                 21                36                     3             36
## 665                14                24                     4             22
## 292                14                20                     3             20
## 180                19                30                     3             15
##     YearsInCurrentRole YearsSinceLastPromotion YearsWithCurrManager
## 541                  9                       9                   11
## 429                  8                       3                    7
## 30                   6                       2                   13
## 665                  6                       5                   17
## 292                  7                       1                    8
## 180                 11                       2                   12
```

Comments: Emma is predicted to stay at the company. All of Emma's 7 nearest neighbours have stayed in the company. They are between 43-55 years of age with 20+ working years in the industry (which seems they are really qualified and have experience). All of them are working with the company for 15+ years with 5+ years in their current role which indicates high job satisfaction.

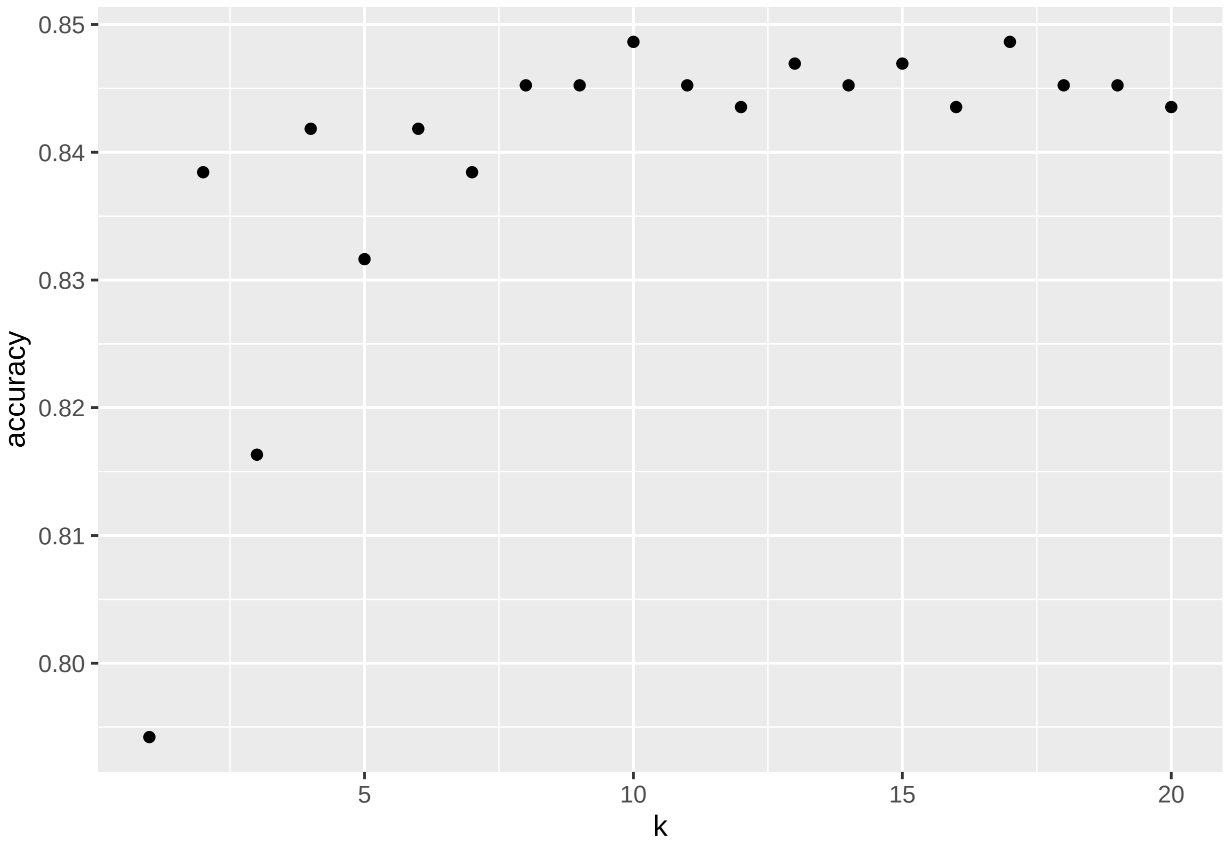
Task 9:Use your validation set to help you determine an optimal k-value

```
accuracy.df <- data.frame(k = seq(1, 20, 1), accuracy = rep(0, 20))

for(i in 1:20) {
  knn.pred <- knn(Train.norm.df[, 2:12], Valid.norm.df[, 2:12],
                  cl = Train.norm.df[, 1], k = i)
  accuracy.df[i, 2] <- confusionMatrix(knn.pred, Valid.norm.df[, 1])$overall[1]
}

accuracy.df
```

```
##      k  accuracy
## 1    1 0.7942177
## 2    2 0.8384354
## 3    3 0.8163265
## 4    4 0.8418367
## 5    5 0.8316327
## 6    6 0.8418367
## 7    7 0.8384354
## 8    8 0.8452381
## 9    9 0.8452381
## 10  10 0.8486395
## 11  11 0.8452381
## 12  12 0.8435374
## 13  13 0.8469388
## 14  14 0.8452381
## 15  15 0.8469388
## 16  16 0.8435374
## 17  17 0.8486395
## 18  18 0.8452381
## 19  19 0.8452381
## 20  20 0.8435374
```

Task 10: Using either the base graphics package or ggplot, make a scatterplot with the various k values that you used on your x-axis, and the accuracy metrics on the y-axis.

```
ggplot(accuracy.df, aes(x = k , y = accuracy)) + geom_point()
```



Task 11: Re-run your knn() function with the optimal k-value that you found previously

Q: What result did you obtain? Was it different from the result you saw when you first ran the k-nn function? Also, what were the outcome classes for each of your person's k-nearest neighbors?

```
KNN <- knn(train = Train.norm.df[, 2:12], test = new.norm.df,
        cl = Train.norm.df[, 1], k = 9)

KNN
```

```
## [1] No
## attr(,"nn.index")
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
## [1,]  541  429   30  415  665  292  180  705  152
## attr(,"nn.dist")
##          [,1]     [,2]     [,3]     [,4]    [,5]     [,6]     [,7]     [,8]
## [1,] 3.909966 4.316404 4.511436 4.570719 4.57336 4.758622 4.766593 4.768481
##          [,9]
## [1,] 4.778889
## Levels: No
```

```
neighbours_new <- Train_KNN[c(541, 429, 30, 665, 292, 180, 705, 152),]
neighbours_new
```

```
##     Attrition Age DistanceFromHome MonthlyIncome NumCompaniesWorked
## 541        No  45                7          5210                  1
## 429        No  53                2         15427                  2
## 30         No  55               26         19586                  1
## 665        No  44                7         10248                  3
## 292        No  43                6          4081                  1
## 180        No  49                6         13966                  2
## 705        No  40                6         16437                  1
## 152        No  51                5         14026                  1
##     PercentSalaryHike TotalWorkingYears TrainingTimesLastYear YearsAtCompany
## 541                18                24                     2             24
## 429                16                31                     3             25
## 30                 21                36                     3             36
## 665                14                24                     4             22
## 292                14                20                     3             20
## 180                19                30                     3             15
## 705                21                21                     2             21
## 152                11                33                     2             33
##     YearsInCurrentRole YearsSinceLastPromotion YearsWithCurrManager
## 541                  9                       9                   11
## 429                  8                       3                    7
## 30                   6                       2                   13
## 665                  6                       5                   17
## 292                  7                       1                    8
## 180                 11                       2                   12
## 705                  7                       7                    7
## 152                  9                       0                   10
```

Comments: Emma is predicted to stay at the company. All of Emma's 9 neihbours are staying at the company. They are between 43-55 years of age with 20+ working years in the industry (which seems they are really qualified and have experience). All of them are working with the company for 15+ years with 5+ years in their current role which indicates high job satisfaction.

Part 2: Naive Bayes

Task 1: Read the file.

```
emp_category <- read.csv("/Users/shimonyagrawal/Desktop/Grad /Summer 2/AD699_Data Mining/RStudio/Assign
```

Task 2: Run the str() function to check the data type for the variables in this dataframe.For any variables that are not currently factors, convert them into factors.

```
str(emp_category)
```

```
## 'data.frame':    1470 obs. of  10 variables:
##  $ Attrition        : chr  "Yes" "No" "Yes" "No" ...
##  $ BusinessTravel   : chr  "Travel_Rarely" "Travel_Frequently" "Travel_Rarely" "Travel_Frequently" .
##  $ Department       : chr  "Sales" "Research & Development" "Research & Development" "Research & Deve
##  $ Education        : chr  "Some College" "High School" "Some College" "Master" ...
##  $ EducationField   : chr  "Life Sciences" "Life Sciences" "Other" "Life Sciences" ...
##  $ Gender           : chr  "Female" "Male" "Male" "Female" ...
##  $ JobSatisfaction  : chr  "Very High" "Low" "High" "High" ...
##  $ MaritalStatus    : chr  "Single" "Married" "Single" "Married" ...
##  $ PerformanceRating: chr  "Good" "Excellent" "Good" "Good" ...
##  $ WorkLifeBalance  : chr  "Bad" "Better" "Better" "Better" ...
```

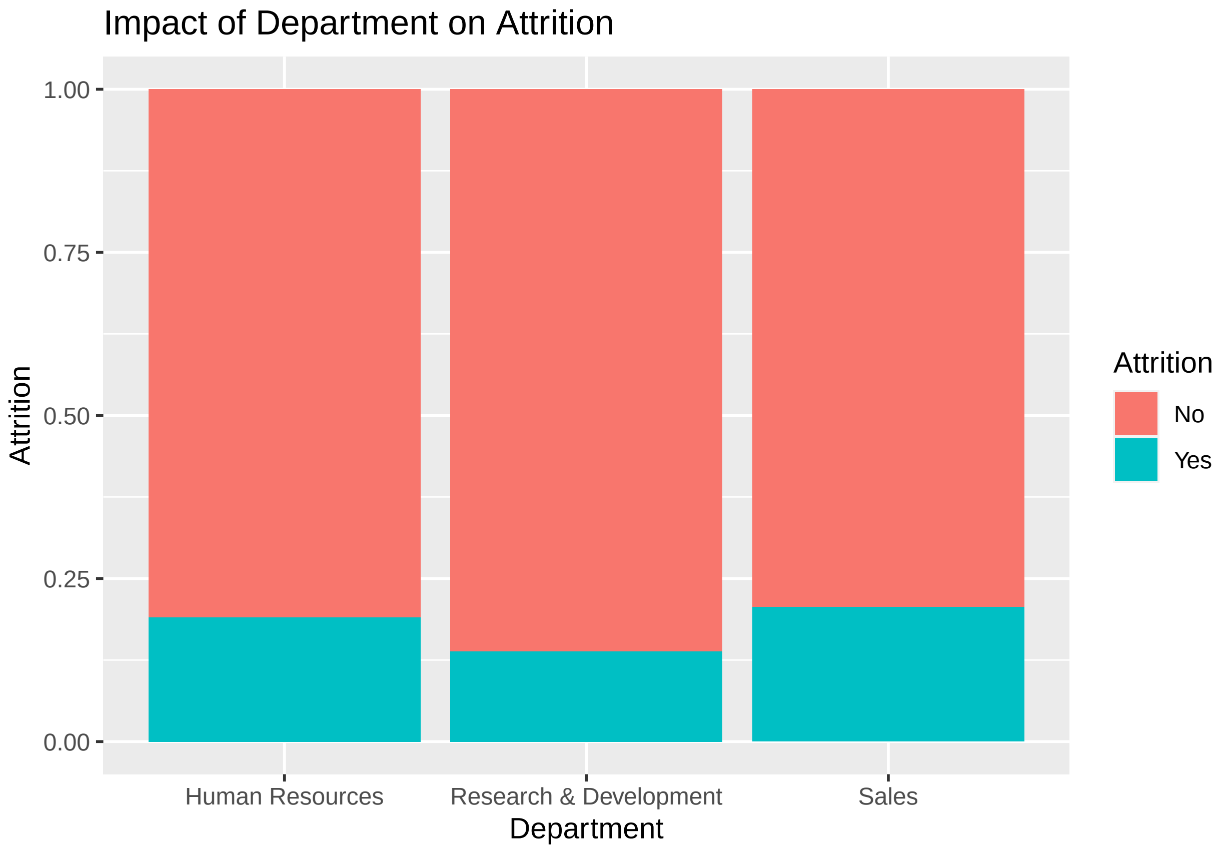
```
emp_category <- emp_category %>%
  mutate_if(is.character, as.factor)

str(emp_category)
```

```
## 'data.frame':    1470 obs. of  10 variables:
##  $ Attrition        : Factor w/ 2 levels "No","Yes": 2 1 2 1 1 1 1 1 1 1 ...
##  $ BusinessTravel   : Factor w/ 3 levels "Non-Travel","Travel_Frequently",..: 3 2 3 2 3 2 3 3 2 3 ..
##  $ Department       : Factor w/ 3 levels "Human Resources",..: 3 2 2 2 2 2 2 2 2 2 ...
##  $ Education        : Factor w/ 5 levels "Bachelor","Doctorate",..: 5 3 5 4 3 5 1 3 1 1 ...
##  $ EducationField   : Factor w/ 6 levels "Human Resources",..: 2 2 5 2 4 2 4 2 2 4 ...
##  $ Gender           : Factor w/ 2 levels "Female","Male": 1 2 2 1 2 2 1 2 2 2 ...
##  $ JobSatisfaction  : Factor w/ 4 levels "High","Low","Very High",..: 3 2 1 1 2 3 4 1 1 1 ...
##  $ MaritalStatus    : Factor w/ 3 levels "Divorced","Married",..: 3 2 3 2 2 3 2 1 3 2 ...
##  $ PerformanceRating: Factor w/ 2 levels "Excellent","Good": 2 1 2 2 2 2 1 1 1 2 ...
##  $ WorkLifeBalance  : Factor w/ 4 levels "Bad","Best","Better",..: 1 3 3 3 3 4 4 3 3 4 ...
```

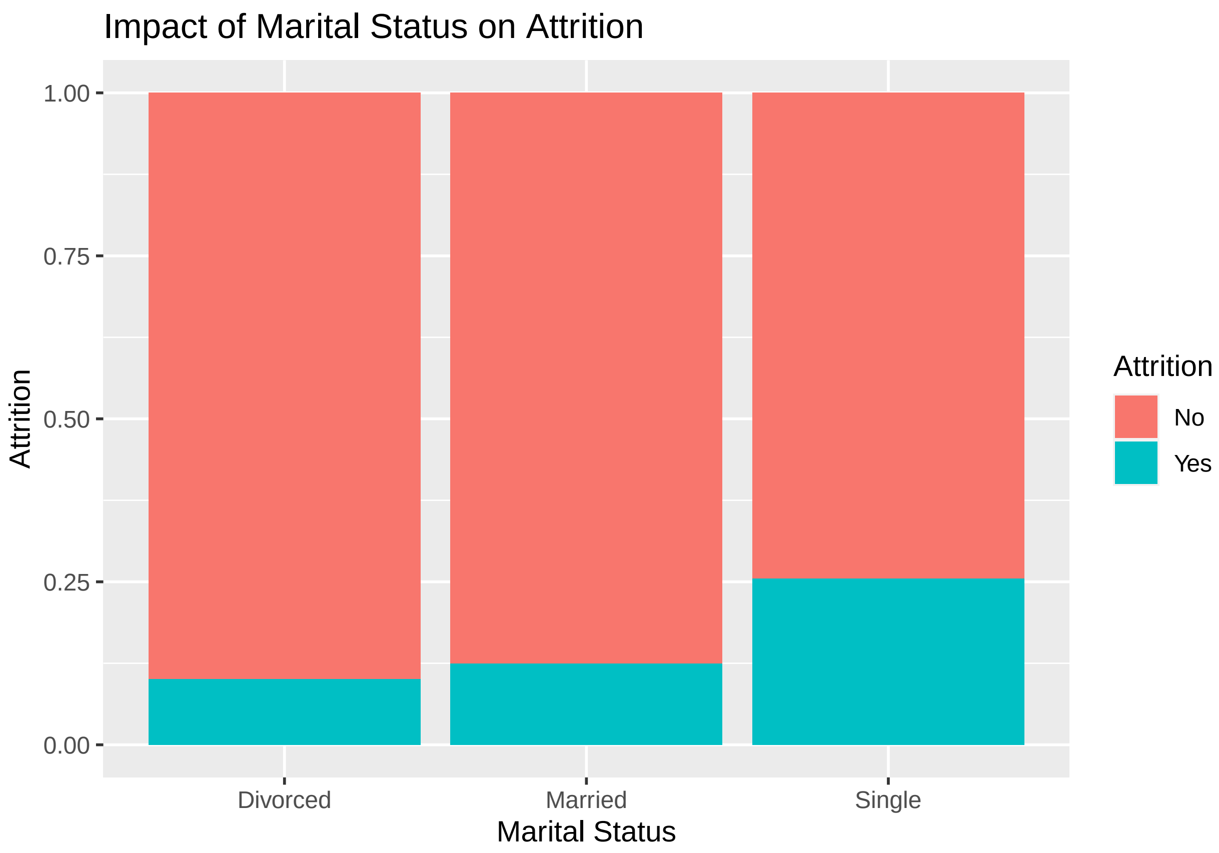
Task 3: Choose any three predictor variables from the dataset. For the three that you chose, make a barplot for each one. Each barplot should show one of your chosen categories on the x-axis, with Attrition as the fill variable. You should build proportional barplots (you can achieve this by adding position="fill" inside your geom layer). You should generate three separate barplots for this step.

Q: Are there any generalizations that you can make about these variables' relationship with attrition? Do some variables look like they'll have more predictive power than others?

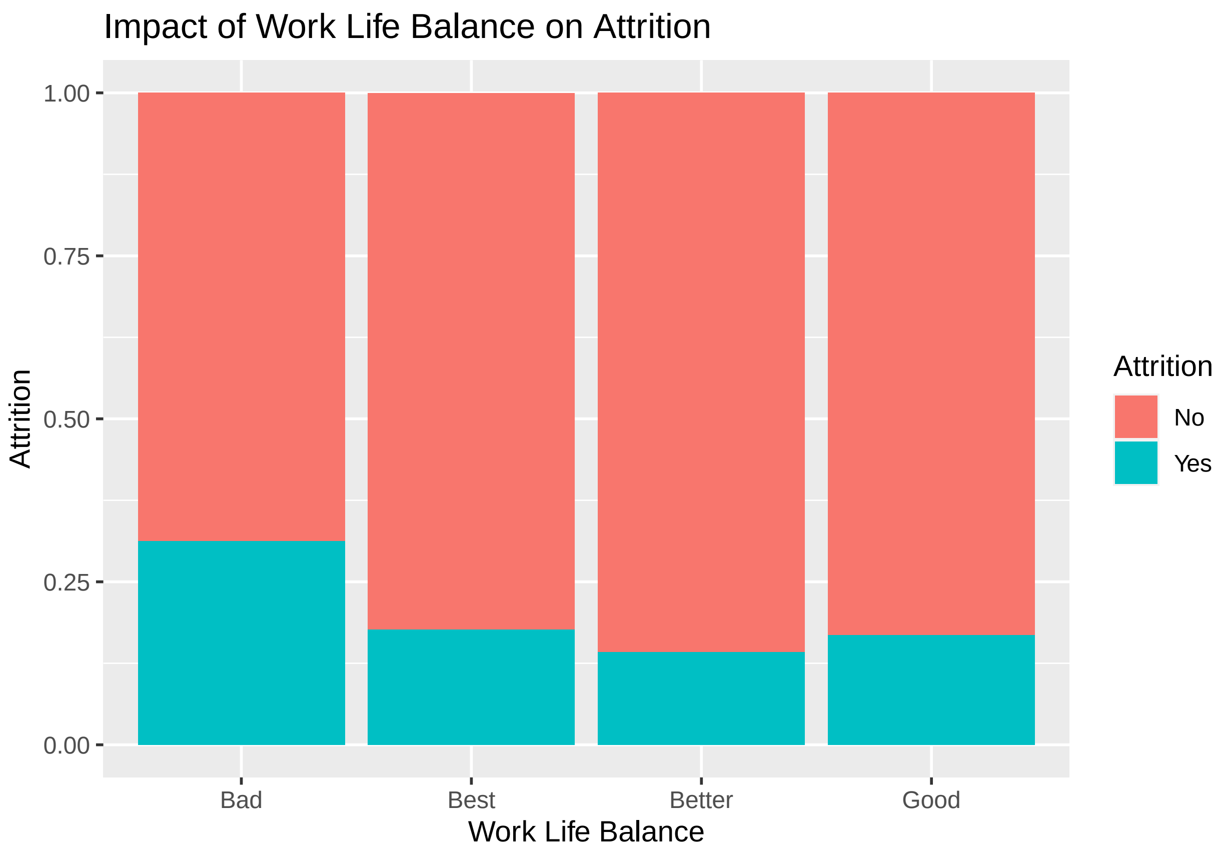
```
# Department

ggplot(emp_category, aes(x = Department, fill = Attrition)) +
  geom_bar(position = 'fill') +
  ggtitle("Impact of Department on Attrition") +
  xlab('Department') +
  ylab('Attrition')
```



```
# Marital Status

ggplot(emp_category, aes(x = MaritalStatus, fill = Attrition)) +
  geom_bar(position = 'fill') +
  ggtitle("Impact of Marital Status on Attrition") +
  xlab('Marital Status') +
  ylab('Attrition')
```



```
# WorkLifeBalance

ggplot(emp_category, aes(x = WorkLifeBalance, fill = Attrition)) +
  geom_bar(position = 'fill') +
  ggtitle("Impact of Work Life Balance on Attrition") +
  xlab('Work Life Balance') +
  ylab('Attrition')
```



Comments: The 3 variables I chose are: Department, Marital Status and WorkLife Balance. I chose the latter two because personally, I feel that your relationships have a significant impact on your professional choices and also, maintaining a balance leads to a happier life. As they say "All work no play makes Jack a dull boy". Graph 1: Impact of Department on Attrition: It can seen that employees in sales and HR have a higher chance of leaving as compared to R&D. This can be due to higher superiority given to R&D in the workplace as compared to other career choices. Graph 2: Impact of Marital Status on Attrition: It can be seen that a higher number of employees who are single have left the company as compared to married / divorced ones. I feel this can be due to the financial stability and responsibility one needs in a marriage. Also, expenses tend to increase post marriage from getting a house, car to family vacations.Single employees on the other hand tend to be more experimental in their career choices before they chose to settle down. Graph 3: Impact of Work Life Balance on Attrition: Employees with bad worklife balance tend to have lef the company as compared to those with best/better/good worklife balance. Nowadays, employees value their personal relationships and lives as much as their professional career. Striking a balance between the two largely depends on the work environment and how much company treats their employees. Maintaining a balance between work and home has proven to have a positive effect on mental health too.

Task 4: Using your seed value (the same one from Assignment #2) , partition your data into training (60%) and validation (40%) sets.

```
nrow(emp_category) *.60
```

```
## [1] 882
```

```
emp_category_sample <- sample_n(emp_category, 1470)
Train_NB <- slice(emp_category_sample, 1:882)
Valid_NB <- slice (emp_category_sample, 883:1470)
```

Task 5: Build a naive bayes model, with the response variable Attrition. Use all of the other variables in your training set as inputs.

```r
empcat_NB <- naiveBayes(Attrition ~ ., data = Train_NB)
empcat_NB
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##        No       Yes
## 0.8446712 0.1553288
##
## Conditional probabilities:
##      BusinessTravel
## Y     Non-Travel Travel_Frequently Travel_Rarely
##   No  0.12080537        0.15302013    0.72617450
##   Yes 0.04379562        0.29927007    0.65693431
##
##      Department
## Y     Human Resources Research & Development      Sales
##   No       0.04295302             0.66711409 0.28993289
##   Yes      0.05109489             0.59124088 0.35766423
##
##      Education
## Y       Bachelor  Doctorate High School     Master Some College
##   No  0.39865772 0.03355705  0.11140940 0.27785235   0.17852349
##   Yes 0.44525547 0.01459854  0.16058394 0.21167883   0.16788321
##
##      EducationField
## Y     Human Resources Life Sciences  Marketing    Medical      Other
##   No       0.01610738    0.41744966 0.09932886 0.32348993 0.05637584
##   Yes      0.04379562    0.34306569 0.17518248 0.27007299 0.05109489
##      EducationField
## Y     Technical Degree
##   No        0.08724832
##   Yes       0.11678832
##
##      Gender
## Y        Female      Male
##   No  0.4013423 0.5986577
##   Yes 0.3430657 0.6569343
##
##      JobSatisfaction
## Y          High       Low Very High  Very Low
##   No  0.3127517 0.1879195 0.3248322 0.1744966
##   Yes 0.3430657 0.1824818 0.1897810 0.2846715
##
##      MaritalStatus
## Y      Divorced   Married    Single
##   No  0.2456376 0.4926174 0.2617450
##   Yes 0.1532847 0.3065693 0.5401460
##
##      PerformanceRating
## Y     Excellent      Good
##   No  0.1449664 0.8550336
##   Yes 0.1532847 0.8467153
##
##      WorkLifeBalance
## Y            Bad       Best     Better       Good
##   No  0.03892617 0.10872483 0.61342282 0.23892617
##   Yes 0.08029197 0.08759124 0.56934307 0.26277372
```

Task 6: Show a confusion matrix that compares the performance of your model against the training data, and another that shows its performance against the validation data.

Q: How did your training set's performance compare with your validation set's performance?

```
# Training data
pred.class1 <- predict(empcat_NB, newdata = Train_NB)
confusionMatrix(pred.class1, Train_NB$Attrition)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  735 123
##        Yes  10  14
##
##                Accuracy : 0.8492
##                  95% CI : (0.8239, 0.8722)
##     No Information Rate : 0.8447
##     P-Value [Acc > NIR] : 0.3761
##
##                   Kappa : 0.1338
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.9866
##             Specificity : 0.1022
##          Pos Pred Value : 0.8566
##          Neg Pred Value : 0.5833
##              Prevalence : 0.8447
##          Detection Rate : 0.8333
##    Detection Prevalence : 0.9728
##       Balanced Accuracy : 0.5444
##
##        'Positive' Class : No
##
```

```
# Validation data
pred.class2 <- predict(empcat_NB, newdata = Valid_NB)
confusionMatrix(pred.class2, Valid_NB$Attrition)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  484  90
##        Yes   4  10
##
##                Accuracy : 0.8401
##                  95% CI : (0.808, 0.8688)
##     No Information Rate : 0.8299
##     P-Value [Acc > NIR] : 0.2756
##
##                   Kappa : 0.1395
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.9918
##             Specificity : 0.1000
##          Pos Pred Value : 0.8432
##          Neg Pred Value : 0.7143
##              Prevalence : 0.8299
##          Detection Rate : 0.8231
##    Detection Prevalence : 0.9762
##       Balanced Accuracy : 0.5459
##
##        'Positive' Class : No
##
```

Comments: For the training set, the accuracy is 84% with true positives to be 98% and true negatives 10%. For the validation set, the accuracy is 84% with true positives to be 99% and true negatives 10%. On the basis of accuracy, both sets have similar performance. The training set has 749 true positives and 133 true negatives whereas the validation set has 494 true positives and 94 true negatives. Here, the training set performances better as it has a higher number correct responses.

Task 7: If you had used the naive rule as an approach to classification, how would you have classified all the records in your training set?

Comments: Naive rule is a baseline to evaluate the performance of complicated classifiers. Naive rule relies solely on the outcome information and excludes predictor variables. Here, the naive rule will assign all records to majority whereas naive bayes will find the records with the same predictor, determine which class they belong to and then assign the class to the most relevant record.

Task 8a: Emily

Task 8b: Create a new dataframe for your person that includes his/her category values.

```
Emily <- data.frame(BusinessTravel = 'Travel_Frequently' ,
                   Department = 'Research & Development',
                   Education = 'Master',
                   EducationField = 'Technical Degree' ,
                   Gender = 'Female' ,
                   JobSatisfaction = 'High',
                   MaritalStatus = 'Single' ,
                   PerformanceRating = 'Good ',
                   WorkLifeBalance = 'Good')
```

Task 8c: Use the predict() function in R to predict whether your person will leave the company.

Q: What outcome did your model predict?

```
pred <- predict (empcat_NB, Emily)
pred
```

```
## [1] No
## Levels: No Yes
```

Comment: The model predicted that Emily is likely to stay at the company.

Task 8d: Use the predict() function in R in a slightly different way to determine the probability that your person will leave the company.

Q: What probability did it assign to your person?

```
pred.prob <- predict(empcat_NB, newdata = Emily, type = "raw") ## predict class membership
pred.class <- predict(empcat_NB, newdata = Emily)

pred.prob
```

```
##             No      Yes
## [1,] 0.591109 0.408891
```

```
pred.class
```

```
## [1] No
## Levels: No Yes
```

Comments: The model predicted that there is 59% chance Emily will stay in the company and a 48% chance that Emily will leave.

Task 8e: Generate a leave_score (Attrition will be Yes) and a stay_score (Attrition will be No). Fictional Instance: What is the leave_score and stay_score for a male employee with a marketing degree who travels frequently, works in sales and has a very high job statisfaction? A-priori Probability- No: 0.8367347 Yes: 0.1632653 Gender - No: 0.5921409 Yes: 0.6388889 Business Travel- No: 0.16260163 Yes: 0.27777778 Marketing- No: 0.10298103 Yes: 0.14583333 Sales- No: 0.30352304 Yes: 0.34027778 JobSatisfaction- No: 0.3238482 Yes: 0.1805556

Q: Use your knowledge of the naive Bayes calculation process to demonstrate how the naive Bayes algorithm generated the probability prediction that you saw in a previous step.

```
stay_score <- 0.8367347 * 0.5921409 * 0.16260163 * 0.10298103 * 0.30352304 * 0.3238482
leave_score <- 0.1632653 * 0.6388889 * 0.27777778 * 0.14583333 * 0.34027778 * 0.1805556

stay_score / (stay_score + leave_score)
```

```
## [1] 0.7585302
```

```
# 0.758 ~ 76% probability that the employee is likely stay
leave_score / (stay_score + leave_score)
```

```
## [1] 0.2414698
```

```
# 0.241 ~ 24% probability that the employee is likely to leave
```

Comments: Based on the results from the emp_NB, I formed a fictional person to demonstrate how the probability prediction works. The model gave probabilities of the employee staying and leaving based on which I generated a stay_score and leave_score.