PU5558-assessment1

Anonymous

Machine learning is subset of Artificial intelligence that allows software application to be more accurate at predicting outcomes without being explicitly programmed to do so. For this machine learning algorithm the following R packages are required.

Load necessary packages

```
library(tidyverse)    # for general data science
library(tidymodels)   # for machine learning
library(kernlab)      # engine for SVM models
library(ggplot2)      # data visualization
library(corrplot)     # for visualising correlation matrices
```

Load chosen dataset

From the provided dataset, the data set chosen for this Machine learning Algorithm is Hip Replacement data set.

```
#read_CSV function is used to upload the data set

Hip_replacement <- read_csv("Hip Replacement CCG 2021 (2).csv")
```

Dataset description

```
nrow(Hip_replacement)
```

```
## [1] 5575
```

```
colnames(Hip_replacement)
```

```
##  [1] "Provider Code"
##  [2] "Procedure"
##  [3] "Revision Flag"
##  [4] "Year"
##  [5] "Age Band"
##  [6] "Gender"
##  [7] "Pre-Op Q Assisted"
##  [8] "Pre-Op Q Assisted By"
```

```
##  [9] "Pre-Op Q Symptom Period"
## [10] "Pre-Op Q Previous Surgery"
## [11] "Pre-Op Q Living Arrangements"
## [12] "Pre-Op Q Disability"
## [13] "Heart Disease"
## [14] "High Bp"
## [15] "Stroke"
## [16] "Circulation"
## [17] "Lung Disease"
## [18] "Diabetes"
## [19] "Kidney Disease"
## [20] "Nervous System"
## [21] "Liver Disease"
## [22] "Cancer"
## [23] "Depression"
## [24] "Arthritis"
## [25] "Pre-Op Q Mobility"
## [26] "Pre-Op Q Self-Care"
## [27] "Pre-Op Q Activity"
## [28] "Pre-Op Q Discomfort"
## [29] "Pre-Op Q Anxiety"
## [30] "Pre-Op Q EQ5D Index Profile"
## [31] "Pre-Op Q EQ5D Index"
## [32] "Post-Op Q Assisted"
## [33] "Post-Op Q Assisted By"
## [34] "Post-Op Q Living Arrangements"
## [35] "Post-Op Q Disability"
## [36] "Post-Op Q Mobility"
## [37] "Post-Op Q Self-Care"
## [38] "Post-Op Q Activity"
## [39] "Post-Op Q Discomfort"
## [40] "Post-Op Q Anxiety"
## [41] "Post-Op Q Satisfaction"
## [42] "Post-Op Q Sucess"
## [43] "Post-Op Q Allergy"
## [44] "Post-Op Q Bleeding"
## [45] "Post-Op Q Wound"
## [46] "Post-Op Q Urine"
## [47] "Post-Op Q Further Surgery"
## [48] "Post-Op Q Readmitted"
## [49] "Post-Op Q EQ5D Index Profile"
## [50] "Post-Op Q EQ5D Index"
## [51] "Hip Replacement EQ5D Index Post-Op Q Predicted"
## [52] "Pre-Op Q EQ VAS"
## [53] "Post-Op Q EQ VAS"
## [54] "Hip Replacement EQ VAS Post-Op Q Predicted"
## [55] "Hip Replacement Pre-Op Q Pain"
## [56] "Hip Replacement Pre-Op Q Sudden Pain"
## [57] "Hip Replacement Pre-Op Q Night Pain"
## [58] "Hip Replacement Pre-Op Q Washing"
## [59] "Hip Replacement Pre-Op Q Transport"
## [60] "Hip Replacement Pre-Op Q Dressing"
## [61] "Hip Replacement Pre-Op Q Shopping"
## [62] "Hip Replacement Pre-Op Q Walking"
```

```
## [63] "Hip Replacement Pre-Op Q Limping"
## [64] "Hip Replacement Pre-Op Q Stairs"
## [65] "Hip Replacement Pre-Op Q Standing"
## [66] "Hip Replacement Pre-Op Q Work"
## [67] "Hip Replacement Pre-Op Q Score"
## [68] "Hip Replacement Post-Op Q Pain"
## [69] "Hip Replacement Post-Op Q Sudden Pain"
## [70] "Hip Replacement Post-Op Q Night Pain"
## [71] "Hip Replacement Post-Op Q Washing"
## [72] "Hip Replacement Post-Op Q Transport"
## [73] "Hip Replacement Post-Op Q Dressing"
## [74] "Hip Replacement Post-Op Q Shopping"
## [75] "Hip Replacement Post-Op Q Walking"
## [76] "Hip Replacement Post-Op Q Limping"
## [77] "Hip Replacement Post-Op Q Stairs"
## [78] "Hip Replacement Post-Op Q Standing"
## [79] "Hip Replacement Post-Op Q Work"
## [80] "Hip Replacement Post-Op Q Score"
## [81] "Hip Replacement OHS Post-Op Q Predicted"
```

The Hip replacement dataset has 81 columns and 5575 rows of data. Most of which are numerical data and few of them are categorical data. Select () is used to select the required column and rest of the column will be dropped off.Selection of columns is totally dependent on the domain knowledge of the dataset and causal relationship with variable bring predicted.

Drop_na () function is used to remove the missing values and unique() will keep only unique rows of the dataset. Glimplse() has been used to have a glimpse of the dataset and it's values. As per data dictionary Age and gender have "*" as missing values. Age and Gender in hip replacement dataset have some missing values. Missing values have been removed by using filter().Summary() has been used to calculate the summary statistics of the dataset i.e. min value, max value, 1st quartile, 3rd Quartile, mean and median of numerical variables.Summary() shows that none of the numerical columns have value "9". As per data dictionary of the dataset "9" is missing value for numerical data. The function also shows that there is no outliers in the data set. str() is used to examine the structure of the dataset.

```
DZ_data <- Hip_replacement %>%
  drop_na() %>%    # remove rows with missing values
  unique()         # keep unique rows


DZ_data <- DZ_data %>%
  select('Age Band','Gender','Post-Op Q EQ5D Index','Pre-Op Q Mobility','Pre-Op Q Activity',
         'Pre-Op Q Discomfort','Pre-Op Q EQ5D Index','Hip Replacement Pre-Op Q Pain',
         'Hip Replacement Pre-Op Q Limping','Hip Replacement Pre-Op Q Stairs',
         'Hip Replacement Pre-Op Q Standing','Hip Replacement Pre-Op Q Work',
         'Hip Replacement Pre-Op Q Score','Pre-Op Q Previous Surgery',
         'Hip Replacement Post-Op Q Pain')  %>%
  rename(Post_op_EQ5D_index = 'Post-Op Q EQ5D Index', Pre_index = 'Pre-Op Q EQ5D Index',
         Pre_op_pain = `Hip Replacement Pre-Op Q Pain`,Limping = `Hip Replacement Pre-Op Q Limping`,
         stairs = `Hip Replacement Pre-Op Q Stairs`, Standing = `Hip Replacement Pre-Op Q Standing`,
         Work = `Hip Replacement Pre-Op Q Work`,Score =`Hip Replacement Pre-Op Q Score`,
         post_operation_pain = `Hip Replacement Post-Op Q Pain`, Age = `Age Band`,
         Mobility = `Pre-Op Q Mobility`, Activity =  `Pre-Op Q Activity`,
         Discomfort = `Pre-Op Q Discomfort`)
```

```
glimpse(DZ_data)
```

```
## Rows: 4,561
## Columns: 15
## $ Age                          <chr> "*", "*", "*", "*", "*", "*", "*", "*", "*~
## $ Gender                       <chr> "*", "*", "*", "*", "*", "*", "*", "*", "*~
## $ Post_op_EQ5D_index           <dbl> 0.727, 0.883, 1.000, 1.000, 0.727, 0.055, ~
## $ Mobility                     <dbl> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, ~
## $ Activity                     <dbl> 3, 3, 3, 2, 3, 2, 2, 2, 2, 2, 3, 2, 2, 2, ~
## $ Discomfort                   <dbl> 3, 3, 3, 3, 2, 2, 3, 2, 3, 2, 2, 2, 3, 3, ~
## $ Pre_index                    <dbl> 0.101, -0.074, -0.003, -0.016, 0.260, 0.62~
## $ Pre_op_pain                  <dbl> 0, 0, 0, 0, 1, 2, 0, 0, 0, 1, 1, 2, 1, 0, ~
## $ Limping                      <dbl> 0, 0, 0, 0, 0, 2, 0, 0, 0, 2, 0, 0, 0, 0, ~
## $ stairs                       <dbl> 3, 2, 2, 2, 2, 4, 2, 1, 2, 3, 2, 2, 3, 2, ~
## $ Standing                     <dbl> 3, 0, 0, 1, 0, 3, 1, 1, 1, 3, 2, 2, 3, 0, ~
## $ Work                         <dbl> 0, 0, 0, 1, 0, 2, 1, 2, 0, 2, 0, 1, 1, 0, ~
## $ Score                        <dbl> 20, 11, 10, 17, 11, 33, 12, 15, 11, 29, 13~
## $ `Pre-Op Q Previous Surgery`  <dbl> 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, ~
## $ post_operation_pain          <dbl> 4, 4, 4, 3, 4, 1, 4, 4, 3, 4, 4, 4, 2, 2, ~
```

```
summary(DZ_data)
```

```
##      Age                Gender           Post_op_EQ5D_index    Mobility
##  Length:4561        Length:4561        Min.   :-0.5940    Min.   :1.000
##  Class :character   Class :character   1st Qu.: 0.6910    1st Qu.:2.000
##  Mode  :character   Mode  :character   Median : 0.8480    Median :2.000
##                                        Mean   : 0.7904    Mean   :1.925
##                                        3rd Qu.: 1.0000    3rd Qu.:2.000
##                                        Max.   : 1.0000    Max.   :3.000
##     Activity       Discomfort     Pre_index        Pre_op_pain
##  Min.   :1.000   Min.   :1.00   Min.   :-0.5940   Min.   :0.000
##  1st Qu.:2.000   1st Qu.:2.00   1st Qu.:-0.0030   1st Qu.:0.000
##  Median :2.000   Median :2.00   Median : 0.2930   Median :0.000
##  Mean   :2.165   Mean   :2.44   Mean   : 0.3259   Mean   :0.531
##  3rd Qu.:2.000   3rd Qu.:3.00   3rd Qu.: 0.6200   3rd Qu.:1.000
##  Max.   :3.000   Max.   :3.00   Max.   : 1.0000   Max.   :4.000
##     Limping           stairs         Standing          Work
##  Min.   :0.0000   Min.   :0.000   Min.   :0.000   Min.   :0.000
##  1st Qu.:0.0000   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.000
##  Median :0.0000   Median :2.000   Median :1.000   Median :1.000
##  Mean   :0.6588   Mean   :1.798   Mean   :1.639   Mean   :1.241
##  3rd Qu.:1.0000   3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:2.000
##  Max.   :4.0000   Max.   :4.000   Max.   :4.000   Max.   :4.000
##      Score       Pre-Op Q Previous Surgery post_operation_pain
##  Min.   : 0.00   Min.   :1.000             Min.   :0.000
##  1st Qu.:11.00   1st Qu.:2.000             1st Qu.:2.000
##  Median :16.00   Median :2.000             Median :3.000
##  Mean   :17.26   Mean   :1.956             Mean   :3.042
##  3rd Qu.:23.00   3rd Qu.:2.000             3rd Qu.:4.000
##  Max.   :48.00   Max.   :9.000             Max.   :4.000
```

```
str(DZ_data)
```

```
## tibble [4,561 x 15] (S3: tbl_df/tbl/data.frame)
##  $ Age                        : chr [1:4561] "*" "*" "*" "*" ...
##  $ Gender                     : chr [1:4561] "*" "*" "*" "*" ...
##  $ Post_op_EQ5D_index         : num [1:4561] 0.727 0.883 1 1 0.727 0.055 0.848 1 0.725 0.656 ...
##  $ Mobility                   : num [1:4561] 2 2 2 2 2 2 2 2 2 2 ...
##  $ Activity                   : num [1:4561] 3 3 3 2 3 2 2 2 2 2 ...
##  $ Discomfort                 : num [1:4561] 3 3 3 3 2 2 3 2 3 2 ...
##  $ Pre_index                  : num [1:4561] 0.101 -0.074 -0.003 -0.016 0.26 0.62 -0.016 0.587 -0.016 0
##  $ Pre_op_pain                : num [1:4561] 0 0 0 0 1 2 0 0 0 1 ...
##  $ Limping                    : num [1:4561] 0 0 0 0 0 0 2 0 0 0 2 ...
##  $ stairs                     : num [1:4561] 3 2 2 2 2 2 4 2 1 2 3 ...
##  $ Standing                   : num [1:4561] 3 0 0 1 0 3 1 1 1 3 ...
##  $ Work                       : num [1:4561] 0 0 0 1 0 2 1 2 0 2 ...
##  $ Score                      : num [1:4561] 20 11 10 17 11 33 12 15 11 29 ...
##  $ Pre-Op Q Previous Surgery: num [1:4561] 1 2 2 2 2 2 2 2 2 2 ...
##  $ post_operation_pain        : num [1:4561] 4 4 4 3 4 1 4 4 3 4 ...
```

Age and Gender are character data in the given dataset. It is first converted into a factor by as.factor() and then to a numeric data using as.numeric() and missing values has also been removed from Age and Gender data.

```
DZ_data <- DZ_data %>%
   filter(Age != "*")

DZ_data <- DZ_data %>%
   filter(Gender != "*")

DZ_data$Age <- as.factor(DZ_data$Age)
DZ_data$Age <- as.numeric(DZ_data$Age)

DZ_data$Gender <-as.numeric(DZ_data$Gender)

summary(DZ_data$Age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   3.000   4.000   3.553   4.000   5.000
```

```
summary(DZ_data$Gender)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00    1.00    2.00    1.66    2.00    2.00
```

```
str(DZ_data$Age)
```

```
##  num [1:2738] 3 3 3 3 3 3 3 3 3 4 ...
```

```
str(DZ_data$Gender)
```

```
##  num [1:2738] 2 2 2 2 2 2 2 2 2 1 ...
```

Suitable machine learning algorithm for three questions:

1. Before the operation, can we estimate the post-operative EQ5D index for a patient?

Answer - estimation refers to the process by which one makes inferences about a population, based on information obtained from a sample. post-operative EQ5D index is a continuous numerical data.Other data of the dataset "DZ_data" are also numerical data. Most suitable and robust machine learning Algorithm to answer this question will be Linear regression supervised Machine Learning Algorithm, given all the condition of Linear Regression are being satisfied as estimation of post-operative EQ5D index will be hypothesized. Conditions of Linear regression are as follows

1. linear relationship between predictors and outcome
2. Independence of residuals
3. Normal distribution of residuals
4. Equal variance of residuals

```
str(DZ_data$Post_op_EQ5D_index)
```

```
##  num [1:2738] 0.552 -0.016 1 0.587 0.62 0.656 1 0.883 1 1 ...
```

2. Before the operation, can we predict how much pain a patient will have after the operation?

Answer - Post operation pain is ordered data with values 0,1,2,3,4.

"0" = "Impossible" "1" = "extreme difficulty" "2" = "moderate difficulty" "3" = "little difficulty" "4" = "Yes, easily"

Since there are only 5 values it can be treated as categories and classification Supervised Machine Learning algorithm Eg SVM or Random Forrest can be used to predict how much pain a patient will have after the operation.

```
table(DZ_data$post_operation_pain)
```

```
##
##    0    1    2    3    4
##  158  258  276  683 1363
```

3. Before the operation, can we calculate how many patients have had previous surgery?

Answer - how many patients have had previous surgery can be calculated before surgery without using Machine learning algorithm using table(). As per data dictionary of the dataset 1 = " Yes", 2 = " No", 9 = "missing value". 377 patients have had previous surgery, 4159 did not have any surgery and for 25 patients, information is mission whether they has surgery or not before.

```
table(DZ_data$`Pre-Op Q Previous Surgery`)
```

```
##
##    1    2    9
##  209 2510   19
```

Model building to answer chosen question

Question 2 has been chosen to build a machine learning model. Question no 2 is as follows

"2. Before the operation, can we predict how much pain a patient will have after the operation?"

Machine learning Model chosen for answering the above question is Support vector machines (SVMs)because of the following reason.

1. The dataset for answering the above question is not very large,It has 2738 rows
2. Features of the dataset are not on the various scales.
3. It has a regularisation parameter, which makes the user think about avoiding over-fitting and performs well when there is a clear indication of separation between classes
4. Support Vector Machines may be relatively sluggish when it comes to training especially with large datasets, however, when it comes to prediction they are quite fast.
5. Most of the dependent variables are numeric and there are no outliers in the dataset

For SVM as classification Supervised Machine Learning Algorithm, the dependent variable should be a factor. as.factor() has been used to convert Post operation pain data into factor.

```
# Converting outcome variable into a factor

DZ_data$post_operation_pain <- as.factor(DZ_data$post_operation_pain)
```

Steps to create a Machine Learning model are as follows

1. Data Splitting
2. Selection and preprocessing of predictors
3. Model specification and training
4. Data splitting

Data splitting is a process in Machine learning where the available data is being partitioned in to two sets, train set and test set. The train dataset is used to develop a predictive model and the test set is used to test the model's performance and accuracy of the predictive model. Usually the data set is divided into 80% training data set and 20% testing set. Split function has been used to split the DZ_data dataset and strata has been defined as "post_operation_pain" so that its distribution in both training and testing dataset is similar. Both dataset has been saved as hip_train and hip_test respectively. hip_train data set which contains 80% of the DZ_data will be used to develop Machine learning model.

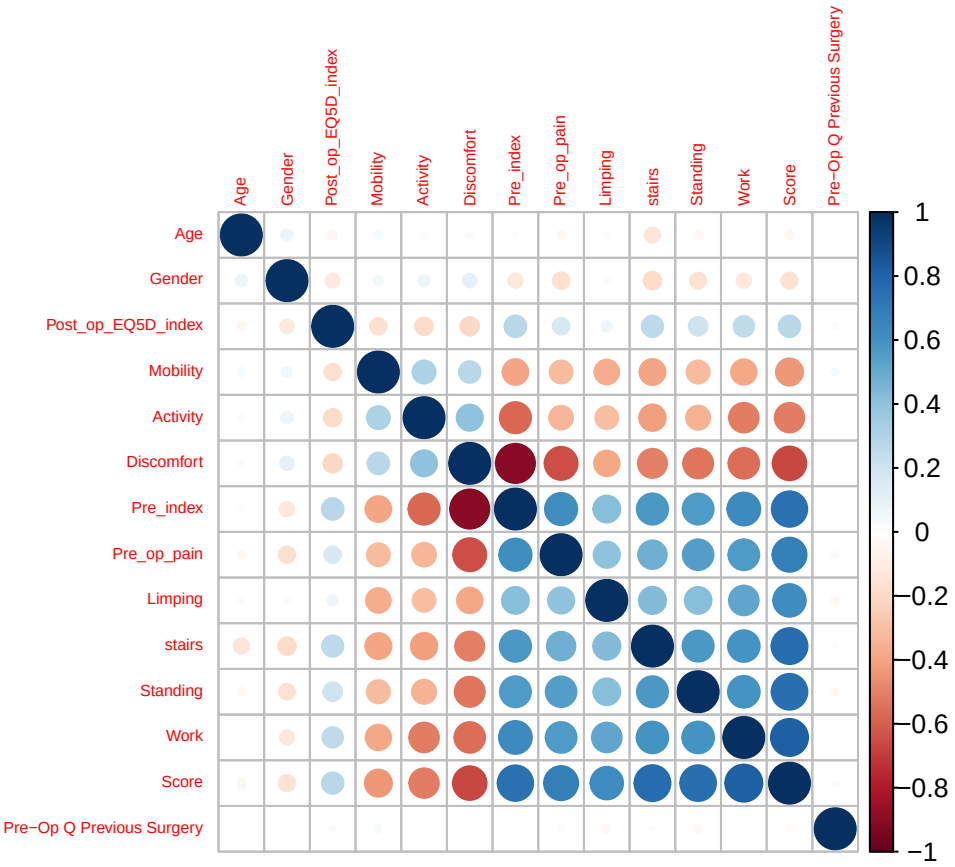
```
# Create the split: hip_split

hip_split <- DZ_data %>%
    initial_split(prop = 0.8,        # proportion in training set
                  strata = post_operation_pain) # ensure distribution of "post_operation_pain"
                                                #in the two datasets is similar
# Save the two datasets as objects: hip_train and hip_test
hip_train <- training(hip_split)
hip_test <- testing(hip_split)
```

2. Selection and preprocessing of predictors

As mentioned above selection of predictors for the machine learning model is dependent upon domain knowledge and causal relationship of the predicting variable. However different graphs can used to visualize relationship between different variable of the dataset and the outcome variable. Cor() can be used to see the correlation between the variables in our dataset. We can see there is positive correlation between post_operation_pain and Score, work and standing and so on

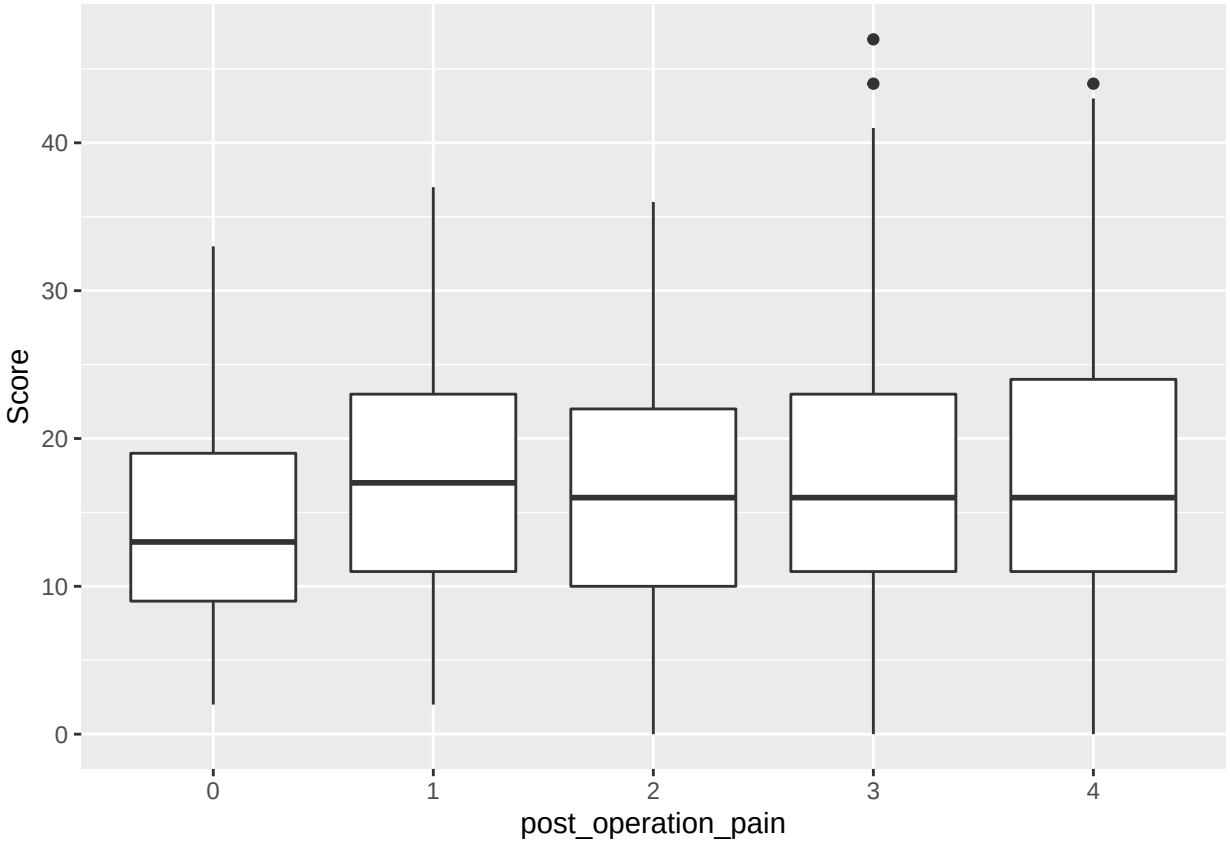
```
 # we can only use numeric variables to calculate correlation
DZ_cor <- cor(hip_train %>% select_if(is.numeric))
corrplot(DZ_cor, tl.cex = 0.5)
```



Box plots can be used to understand relationship between continuous predictors "score" and outcome "post_operation_pain".Below boxplot shows the distribution of "score" is almost symmetric with little bit of skewness in value "3" and "4". As per the data dictionary provided Post_operation_pain values are as follows

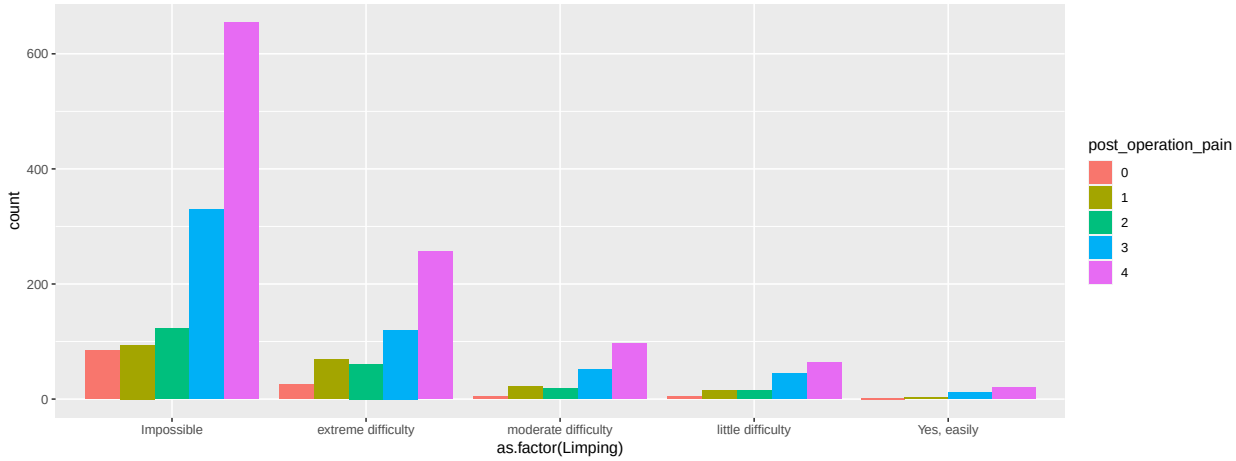
0 = "sever" 1 = "moderate" 2 = "mild" 3 = "very mild" 4 = "none"

```
hip_train %>%
  ggplot(aes(x = post_operation_pain, y = Score)) +
  geom_boxplot()
```



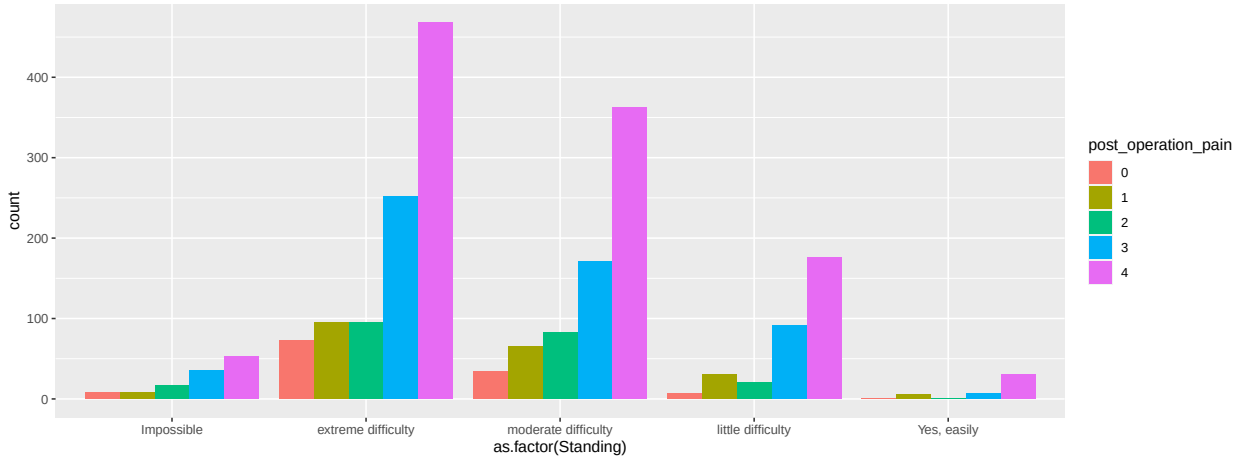
Limping,Standing,Stairs and age Band are all ordinal data with only five possible values. Hence Ordinal data can be treated as categorical data. A bar plot can be plotted to bring more meaning to understand the relationship between predictor and outcome variable.

```
hip_train %>%
  ggplot(aes(x = as.factor(Limping), fill = post_operation_pain)) +
  geom_bar(position="dodge")+
  scale_x_discrete(labels=c("0" = "Impossible",
                            "1" = "extreme difficulty",
                            "2" = "moderate difficulty",
                            "3" = "little difficulty",
                            "4" = "Yes, easily"))
```



From the graph for Limping and post operation pain, there is a clear difference between "Impossible" and other categories like extreme, moderate and little difficulty. Feature engineering could be done, where a new binary variable can be created one for "impossible" and another one combining rest of the categories.

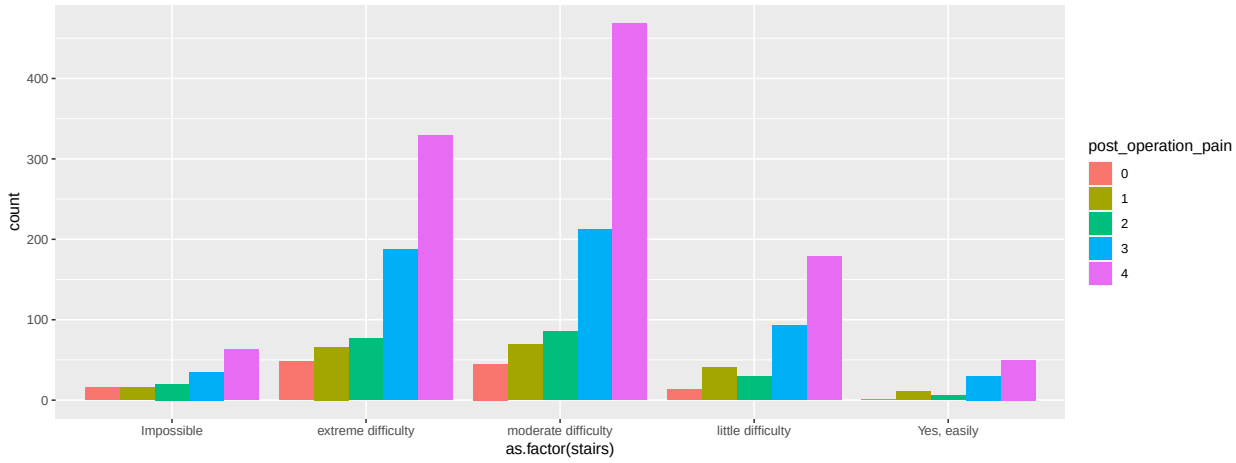
```
hip_train %>%
  ggplot(aes(x = as.factor(Standing), fill = post_operation_pain)) +
  geom_bar(position="dodge")+
  scale_x_discrete(labels=c("0" = "Impossible",
                            "1" = "extreme difficulty",
                            "2" = "moderate difficulty",
                            "3" = "little difficulty",
                            "4" = "Yes, easily"))
```



In this graph "extreme difficulty" and "moderate difficult" has most of the values compared to rest of the categories.

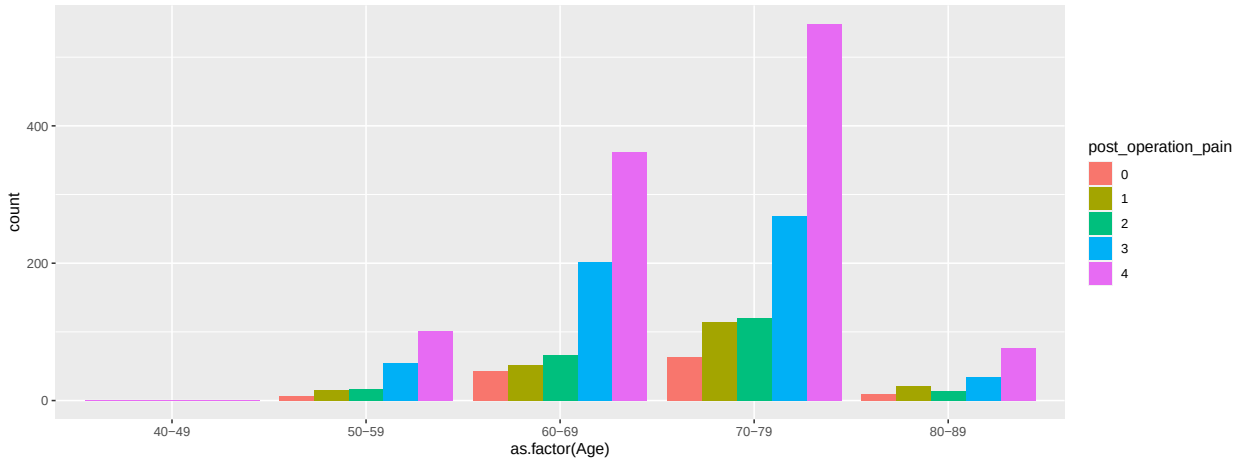
```
hip_train %>%
  ggplot(aes(x = as.factor(stairs), fill = post_operation_pain)) +
  geom_bar(position="dodge")+
  scale_x_discrete(labels=c("0" = "Impossible",
```

```
                          "1" = "extreme difficulty",
                          "2" = "moderate difficulty",
                          "3" = "little difficulty",
                          "4" = "Yes, easily"))
```



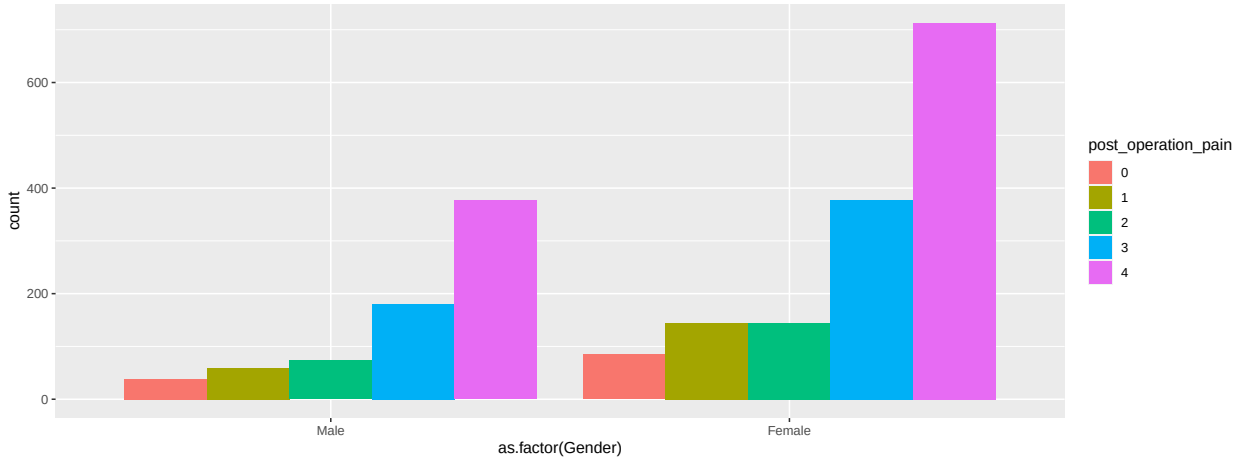
Grapgh for "stairs" is almost similar to "standing"

```
hip_train %>%
  ggplot(aes(x = as.factor(Age), fill = post_operation_pain)) +
  geom_bar(position="dodge")+
  scale_x_discrete(labels=c("1" = "40-49",
                            "2" = "50-59",
                            "3" = "60-69",
                            "4" = "70-79",
                            "5" = "80-89"))
```



From the graph patient within the age group "60-69" and "70 to 79" have more concentration of data for post operation pain.

```
hip_train %>%
  ggplot(aes(x = as.factor(Gender), fill = post_operation_pain)) +
  geom_bar(position="dodge")+
  scale_x_discrete(labels=c("1" = "Male", "2" = "Female"))
```



From the gender graph, female will have more post operative pain as compared to male.

A data set can have bias and variances. Bias skews the result of an algorithm in favor or against the output. It is a systematic error which occurs due to incorrect assumptions during the Machine learning process, whereas Variance refers to the changes in the model when a different portion of training dataset is chosen. Bias and variance are inversely connected. It is important to have right balance of bias and variance in the dataset. Goal is to keep bias as low as possible while introducing acceptable levels of variances which can be achieved by increasing the complexity or increasing the training data set.

Recipe() is applied to use variable age, gender, score, mobility, activity, discomfort, limping, Standing, stairs from the training dataset. Step_normalization function is used to give equal importance to each variable. Normalization would stop any variable to steer the model performance in one direction because of being a bigger number. Step_zv() is used to remove variable that contains only one single value. Step_corr() is used to remove the variable that have large absolute correlation with other variables.

```
simple_rec <- hip_train %>%
  recipe(post_operation_pain ~ Age + Gender + Score + Mobility + Activity + Discomfort + Limping + Stan
  #normalize numeric data to have a standard deviation of one and a mean of zero.
  step_normalize(all_predictors()) %>%
   #creates specification of a recipe that removes variables that contain only a single value
  step_zv(all_predictors()) %>%
   #this will remove variables that have large absolute correlations with other variables.
  step_corr(all_predictors())
```

3. Model specification and training

Model specification in SVM can be of two type

1. Linear SVM model specification
2. Radial basis function Model Specification

Linear SVM Model Specification - It is a linear model for classification and regression problems. The algorithm creates a line or hyperplane which separates data into classes.

Radial basis function Model Specification - It is a function that assigns a real value to each input from its domain.

Cost - A positive number for the cost of predicting a sample within or on the wrong side of the margin.

To specify the model, we need to make three decisions:

1. Model type (for both Linear and radial basis function)
2. Model Mode
3. Model engine

Mode and engine are defined by set_mode() and set_engine() and keeping the cost value at default i.e. 1. Since predicting data is ordinal with only five values, it can be used as categorical data. Hence mode is defined as "classification" and engine as "Kernlab" for both of the specification.

```
# linear SVM model specification
linear_svm_spec <- svm_linear(cost = 1) %>%
  set_mode("classification") %>%
  set_engine("kernlab")


# radial basis function SVM model specification
rbf_svm_spec <- svm_rbf(cost = 1) %>%
  set_mode("classification") %>%
  set_engine("kernlab")
```

Workflow

"workflow" pairs a recipe and a model specification, so that it can be used together. One workflow has been build up for each of the models.

```
# Linear SVM workflow: heart_liner_wflow

hip_liner_wflow <-workflow() %>%
           add_recipe(simple_rec) %>%
           add_model(linear_svm_spec)


# RBF SVM workflow: heart_rbf_wflow


hip_rbf_wflow <-workflow() %>%
           add_recipe(simple_rec) %>%
           add_model(rbf_svm_spec)
```

Model training

Model is now ready to fit using workflows. Bootstrap resampling has been used. Bootstrap Sampling is a method that involves drawing of sample data repeatedly with replacement from a data source to estimate a population parameter. It commonly used to reduce variance within a noisy dataset. Bootstarp sampling is

done on hip_train dataset. Both of the workflows has been used using fit_resamples() for both linear SVM model and radial basis model.

```
# Create bootstraps: hip_boot

hip_boot <- bootstraps(hip_train)  # create bootstraps

# Fit the linear SVM model: hip_linear_wflow_fit



hip_linear_wflow_fit <- hip_liner_wflow %>%
  fit_resamples(
    resamples = hip_boot,
    control = control_resamples(save_pred = TRUE)
  )
```

```
## Warning: package 'rlang' was built under R version 4.1.3
```

```
# Fit the RBF SVM model: hip_rbf_wflow_fit

hip_rbf_wflow_fit <- hip_rbf_wflow %>%
  fit_resamples(
    resamples = hip_boot,
    control = control_resamples(save_pred = TRUE)
  )
```

Model evaluation

performance metrics can be calculated using collect_metrics().

```
collect_metrics(hip_linear_wflow_fit)
```

```
## # A tibble: 2 x 6
##   .metric  .estimator  mean     n std_err .config
##   <chr>    <chr>      <dbl> <int>   <dbl> <chr>
## 1 accuracy multiclass 0.501    25 0.00232 Preprocessor1_Model1
## 2 roc_auc  hand_till  0.502    25 0.00275 Preprocessor1_Model1
```

```
collect_metrics(hip_rbf_wflow_fit)
```

```
## # A tibble: 2 x 6
##   .metric  .estimator  mean     n std_err .config
##   <chr>    <chr>      <dbl> <int>   <dbl> <chr>
## 1 accuracy multiclass 0.493    25 0.00243 Preprocessor1_Model1
## 2 roc_auc  hand_till  0.503    25 0.00324 Preprocessor1_Model1
```

Both linear model and the RBF model have almost equal accuracy. Both of the model's accuracy is around 0.5. Accuracy depend upon the quality of dataset used to train the algorithm. Accuracy can also be improved by selecting a more suitable predictor and number of suitable predictor variable which can come from domain

knowledge of the data set. It can also be improved by adjusting cost value, Feature engineering and by tuning hyperparameters

A confusion matrix can be calculated for both of the SVM models.A confusion matrix is a table which helps in visualizing the performances of an algorithm.

Confusion matrix for Linear model is as follows

```
results <-  hip_linear_wflow_fit %>%
  collect_predictions()

results %>%
  conf_mat(post_operation_pain, .pred_class)
```

```
##            Truth
## Prediction     0     1     2     3     4
##          0     0     0     0     0     0
##          1     0     0     0     0     0
##          2     0     0     0     0     0
##          3     0     0     0     0     0
##          4  1119  1859  1962  5062 10031
```

The above confusion matrix shows that the linear model can only predict pain score of "4" correctly and rest are zeros. as the rest of the diagonal value is zeros. However the number of true positive for pain score of 4 is high which is 9974.

Confusion matrix for Linear model is as follows

```
results <-  hip_rbf_wflow_fit %>%
  collect_predictions()

results %>%
  conf_mat(post_operation_pain, .pred_class)
```

```
##            Truth
## Prediction    0    1    2    3    4
##          0    0    0    1    2    0
##          1    0    1    0    0   10
##          2    1    1    0    7    4
##          3   15   26   35   82  221
##          4 1103 1831 1926 4971 9796
```

The confusion matrix for SVM is shown as above. From the matrix we can say that the model can predict pain score of "4" , "3" and "2" correctly as they lie on the diagonal of the confusion matrix.

Confusion matrix of RBF SVM model looks a little better than the confusion matrix of Linear SVM. Hence Radial basis function SVM model is more suitable for the give dataset.

model performance

Now let's look at our testing set. We will use final_fit() on the data split object, which fits on the training data and evaluates on the testing data in one step. The accuracy of the rbf SVM model is 49%.

```r
# Fit the final model
hip_rbf_finalfit <- hip_rbf_wflow %>%
    last_fit(hip_split)

# Print performance metrics on testing data
collect_metrics(hip_rbf_finalfit)
```

```
## # A tibble: 2 x 4
##   .metric  .estimator .estimate .config
##   <chr>    <chr>          <dbl> <chr>
## 1 accuracy multiclass     0.499 Preprocessor1_Model1
## 2 roc_auc  hand_till      0.507 Preprocessor1_Model1
```

Limitations of machine learning model

Machine learning is a process of analyzing data for training models. It can review large volumes of data and discover specific trends and patterns which is otherwise difficult. For Achieving this, machine learning requires massive data set to train on and enough time to let algorithm learn and develop with a considerable amount of accuracy and relevancy.

Machine learning algorithm used above is support vector machine which is a supervised machine learning model that can be used for both classification and Regression algorithms. Although accuracy of SVM models are high , it cannot be used with very big dataset. IT tends to be unusable beyond 20000 data points. The training time of a SVM model is much longer as it's much more computationally intensive.

Choosing an appropriate Kernel function (to handle the non-linear data) is not an easy task. It could be tricky and complex.Algorithmic complexity and memory requirements of SVM are very high. It requires a lot of memory since it stores all the support vectors in the memory and this number grows abruptly with the training dataset size.

feature scaling through standardization which involves rescaling the features such that they have the properties of a standard normal distribution with a mean of zero and a standard deviation of one. It is important preprocessing step for SVM

SVM takes a long training time on large datasets and also it is difficult to understand and interpret by human beings

Although there are limitations of using Machine Learning, it can be incredibly powerful when used in the right ways and in the right places