(1) College students completing a preliminary year

The data for college students was very small but it had a lot of columns which made it very rich. There were 6 character columns and 28 numerical columns. Most of the numerical columns had the value of either 1 or 0

As always, I printed the head of the data and then the summary and immediately noticed the missing values. Because we had character variables, there was a possibility of missing values in these columns too so I checked all the character variables separately. In this process, I found that 4 out of the 6 character variables had 2 rows when all of them were missing

There were several missing values in the remaining 2 columns. To solve the missing values in character columns, we delete the two rows with missing values in 4 columns and the 2 columns with a lot of missing values

The next step was to solve the missing values in numerical columns which is easy. As a simple step, they have been imputed with mean of the data

Then we set a seed so that the results can be replicated. We have divided our data into 85%-15% train test split and then fitted out logistic regression model on the train data only.

The model summary shows us that there were no anomalies in model fitting and the model looks good at the onset. To verify this, we make predictions on the test data and check how many of them were correct. The model accuracy on test data is as follows:

```
predictions  0  1
          0  5  2
          1  3 11
```

This shows that we were able to make 5 out of 8 correct predictions as 0 and 11 out of 13 correct predictions as 1. The total accuracy is 5+11 = 16 correct predictions out of 21. This is equal to 16/21 or 76.19% accuracy and is a great number

We then try to find out the features which decide the model predictions. These features in their order of importance are:

Receptivity.to.Academic.Assistance..percentile.score.before.start.of.semester.	1.18E-04

Completed.Connect...1.yes..0.no.1	6.41E-05
Number.of.Peer.Mentor.Meetings.Attended	6.37E-05
Peer.MentorFrancess	5.55E-05
Peer.MentorCorrinn	5.36E-05
Completed.Connect...1.yes..0.no.0, contract for fall	5.27E-05
Peer.MentorJoli	4.93E-05
Peer.MentorGeorge	4.89E-05
Attended.Orientation...1.yes..0.no.	3.87E-05
Dropout.Proneness..percentile.score.before.start.of.semester.	3.77E-05
Receptivity.to.Academic.Assistance..percentile.score.before.start.of.semester.	1.18E-04

CODE:

```
#Import libraries
library(dplyr)
library(caret)

#Read data
data = read.csv("232408682483120_File.csv")

#View data
head(data)
summary(data) #missing value alert

#Check character columns
unique(data$Gender) #So there are missing values here too

#Check how many missing in gender
length(which(data$Gender=="")) #Only 2. Delete
data = data %>% filter(!Gender=="")

#Check how many missing in Federal Ethnic Group
length(which(data$Federal.Ethnic.Group=="")) #No missing values now
#Check how many missing in Peer Mentor
length(which(data$Peer.Mentor=="")) #No missing values now
#Check how many missing in Completed.Connect...1.yes..0.no.
length(which(data$Completed.Connect...1.yes..0.no.=="")) #No missing values
now
#Check how many missing in Reason.for.not.Completing.Connect
length(which(data$Reason.for.not.Completing.Connect=="")) #Should be removed
#Check how many missing in Reason.not.Retained
```

```
length(which(data$Reason.not.Retained=="")) #Should be removed

data$Reason.for.not.Completing.Connect=NULL
data$Reason.not.Retained=NULL

#Let's fill all the remaining missing values by their mean
for(i in 1:ncol(data)){
  data[is.na(data[,i]), i] <- mean(data[,i], na.rm = TRUE)
}

#Divide data in train and test
set.seed(5)

random_rows = sample(c(TRUE, FALSE), nrow(data), replace=TRUE,
prob=c(0.85,0.15))
college_train = data[random_rows, ]
college_test = data[!random_rows, ]

#Train the model
model.fit = glm(Retained.F17.F18...1.yes..0.no. ~ .,  data =
college_train,  family = "binomial")
#Check how the model performed
model.fit #Some variables have NA values. They don't have any value

#Model accuracy check
predictions = predict(model.fit, college_test, type = "response")
predictions = round(predictions,0)
table(predictions,college_test$Retained.F17.F18...1.yes..0.no.) #16/21 =
76.19%

#Check for important features
list_of_feature_importance = as.data.frame(varImp(model.fit))
list_of_feature_importance = list_of_feature_importance %>%
arrange(desc(Overall) )
head(list_of_feature_importance,10)
```

(2) Heart Health Data

The data for heart health can be used in various ways because the delay in number of days which the patient makes before needing medical attention is initially a continuous variable and it has to be converted into a binary variable using various thresholds. In this assignment, we will look at three different models built from binary variables derived out of ‘delaydays’ variable.

As always, I printed the head of the data and then the summary. The first thing to notice was the presence of ID column which we can delete. The next thing was the presence of 3 outliers in the

delaydays variable which had to be fixed. Along with that there were 2 missing values in livewith variable. We have replaced all the missing values with the mean of data

Part 1 - Predict whether a person seeks medical treatment in 2 days or less

The first model divides the data based on the median and hence it equally distributes the delaydays into 1 and 0. After creating the binary variable, we set a seed so that the results can be replicated. We have divided our data into 85%-15% train test split and then fitted out logistic regression model on the train data only.

The model summary shows us that there were no anomalies in model fitting and the model looks good at the onset. To verify this, we make predictions on the test data and check how many of them were correct. The model accuracy on test data is as follows:

```
predictions  0    1
          0  17  17
          1  9   18
```

This shows that we were able to make 17 out of 26 correct predictions as 0 and 18 out of 35 correct predictions as 1. The total accuracy is 17+18 = 35 correct predictions out of 61. This is equal to 35/61 or 57.38% accuracy and is average

We then try to find out the features which decide the model predictions. These features in their order of importance are:

cough	1.941344
edema	1.70179
weightgain	1.68214
DOE	1.495624
PND	1.222872
palpitations	1.182458
Gender	1.128875
nausea	0.778395
chestpain	0.773945
Education	0.757391

This shows that Cough, Edema and Weight gain are the three most important features in people who need medical attention in 2 days or less. It is also interesting that gender and education are also two of the top 10 features. This might mean that people of a particular gender need or seek early medical attention or people with a particular education background need or seek early medical attention

Part 2 - Predict whether a person seeks medical treatment on or less than cohort average

The second model divides the data based on the mean which is relatively larger and is equal to 5.725779. This shows that some values of delaydays are very large and may be affecting the output. Because of the large threshold, we have more than 2/3 of data as 1 and rest as 0. After creating the

binary variable, we set a seed so that the results can be replicated. We have divided our data into 85%-15% train test split and then fitted out logistic regression model on the train data only.

However, when we try to make predictions from the second model then we realize that the model is dumb because it is predicting everything as 1. This is primarily due to class imbalance stemming from very high values in delaydays column. The model accuracy on test data is as follows:

```
predictions  0    1
          0  0    1
          1  16   44
```

This shows that we were able to make 0 out of 16 correct predictions as 0 and 44 out of 45 correct predictions as 1. This is equal to 44/61 or 72.13% accuracy

We then try to find out the features which decide the model predictions. These features in their order of importance are:

edema	2.427582
nausea	1.946682
PND	1.167519
DOE	1.15639
fatigue	1.144802
weightgain	1.101344
Gender	1.045032
Education	0.88796
orthopnea	0.853992
tightshoes	0.816436

This shows that Edema, Nausea and PND are the three most important features in people who need medical attention in 5.7 days or less. It is also interesting that gender and education are still two of the top 10 features. Another interesting thing is the presence of tightshoes in the top 10 features. Compared to the previous model, cough, Palpitations and chest pain are missing. The first model had cough as the most important feature whereas it is not in the top 10 in the second model

Part 3 - Predict whether a person seeks medical treatment in less than or equal to 1 days

The last model divides the data based on the 1 day or less which is less than the mean and this time we have more than 2/3 of data as 0. After creating the binary variable, we set a seed so that the results can be replicated. We have divided our data into 85%-15% train test split and then fitted out logistic regression model on the train data only.

Unlike Part 2, the model does have some prediction accuracy. The model accuracy on test data is as follows:

```
predictions  0    1
          0  35   17
          1  3    6
```

This shows that we were able to make 35 out of 38 correct predictions as 0 and 6 out of 20 correct predictions as 1. This is equal to 35+6= 41/61 or 67.21% accuracy

We then try to find out the features which decide the model predictions. These features in their order of importance are:

edema	2.972368
cough	2.301375
orthopnea	2.113963
DOE	1.597461
Education	1.320676
Livewith	1.256441
weightgain	1.143784
fatigue	1.110786
PND	1.079065
tightshoes	0.891531

This time, we have some features from both of the previous models. Cough comes back as important feature in model 1 and 3 and tightshoes which was important in model 2 and model 3. Edema, Cough and Orthopnea are the three most important features and orthopnea is unique to this situation. This means that if a person's status of orthopnea is known, then it is very easy to determine whether the person will need medical attention in 1 day or less. In this model, gender does not matter

CODE:

```
#Import libraries
library(dplyr)
library(caret)
library(readxl)

data = read_excel("789968165805283_File.xls")

#View data
head(data) #Why is ID here?
summary(data) #Missing values

#Remove ID
data$ID=NULL

#Let's fill all the remaining missing values by their mean
for(i in 1:ncol(data)){
  data[is.na(data[,i]), i] = mean(as.matrix(data[,i]), na.rm = TRUE)
}

#Part 1 - Predict whether a person seeks medical treatment in 2 days or less
median(data$delaydays) #2.02
data$target_feat = ifelse(data$delaydays<=2, 1,0)
```

```r
table(data$target_feat) #203 1's and 203 0's

#Divide data in train and test
set.seed(5)

random_rows = sample(c(TRUE, FALSE), nrow(data), replace=TRUE,
prob=c(0.85,0.15))
health_train = data[random_rows, ]
health_test = data[!random_rows, ]
health_train$delaydays=NULL

#Train the model
model.fit = glm(target_feat ~ .,  data = health_train,  family = "binomial")
#Check how the model performed
model.fit #Some variables have NA values. They don't have any value

#Model accuracy check
predictions = predict(model.fit, health_test, type = "response")
predictions = round(predictions,0)
table(predictions,health_test$target_feat) #35/61 = 57.38%

#Check for important features
list_of_feature_importance = as.data.frame(varImp(model.fit))
list_of_feature_importance = list_of_feature_importance %>%
arrange(desc(Overall) )
head(list_of_feature_importance,10)

#Part 2 - Predict whether a person seeks medical treatment on or less than
cohort average
avg = mean(data$delaydays,na.rm = TRUE)
avg #5.725779
data$target_feat = ifelse(data$delaydays<=avg, 1,0)
table(data$target_feat) #285 1's and 121 0's

#Divide data in train and test
set.seed(5)

random_rows = sample(c(TRUE, FALSE), nrow(data), replace=TRUE,
prob=c(0.85,0.15))
health_train = data[random_rows, ]
health_test = data[!random_rows, ]
health_train$delaydays=NULL

#Train the model
model.fit = glm(target_feat ~ .,  data = health_train,  family = "binomial")
#Check how the model performed
model.fit #Some variables have NA values. They don't have any value
```

```r
#Model accuracy check
predictions = predict(model.fit, health_test, type = "response")
predictions = round(predictions,0)
table(predictions,health_test$target_feat) #44/61 = 72.13%
#This model is predicting everything as 1. This is a dumb model

#Check for important features
list_of_feature_importance = as.data.frame(varImp(model.fit))
list_of_feature_importance = list_of_feature_importance %>%
arrange(desc(Overall) )
head(list_of_feature_importance,10)

#Part 3 - Predict whether a person seeks medical treatment in less than or
equal to 1 days
data$target_feat = ifelse(data$delaydays<=1, 1,0)
table(data$target_feat) #137 1's and 269 0's

#Divide data in train and test
set.seed(5)

random_rows = sample(c(TRUE, FALSE), nrow(data), replace=TRUE,
prob=c(0.85,0.15))
health_train = data[random_rows, ]
health_test = data[!random_rows, ]
health_train$delaydays=NULL

#Train the model
model.fit = glm(target_feat ~ .,  data = health_train,  family = "binomial")
#Check how the model performed
model.fit #Some variables have NA values. They don't have any value

#Model accuracy check
predictions = predict(model.fit, health_test, type = "response")
predictions = round(predictions,0)
table(predictions,health_test$target_feat) #41/61 = 67.21%

#Check for important features
list_of_feature_importance = as.data.frame(varImp(model.fit))
list_of_feature_importance = list_of_feature_importance %>%
arrange(desc(Overall) )
head(list_of_feature_importance,10)
```