The data is then read in from a CSV file using the read_csv function from the readr
package.
• The data is checked for missing values using the sapply function, and unnecessary
columns are removed using the select function from the dplyr package. The last two
rows are also removed using the slice function from the dplyr package.
• Next, numerical null values are imputed with the mean using the mutate_all function from
the dplyr package.
• Numerical variables are then scaled using the scale function, and outliers are detected
and removed using a custom function that utilizes the quantile and interquartile range.
• The caret package is loaded, and categorical variables are converted to numerical using
the dummyVars and predict functions.
• A correlation matrix is then created using the cor function, and highly correlated columns
are identified using the which and abs functions. These highly correlated columns are
removed from the dataset.
• The data is then split into training and testing sets using the createDataPartition function
from the caret package.
• The logistic regression model is then fit on the training data using the cv.glmnet function
from the glmnet package.
• Predictions are made on the testing data using the predict function, and the accuracy of
the model is calculated. The F1 score is also calculated using the F1_Score function
from the MLmetrics package.
• The coefficients from the fitted model are extracted using the coef function, and a data
frame is created with the variable names and coefficients.
• This data frame is then sorted by absolute value, and the top 10 features are selected.
The top 10 features are then used to fit another logistic regression model on the training
data and make predictions on the testing data.
• The accuracy of this model is also calculated and the value comes out to be 0.9032258
and F1 score is 0.8888889
• It can be observed that selecting the top 10 features and training on them doesn't affect
our accuracy by a significant amount, hence all the features are significant to our model
and should be considered while training.

Code:

```
# Installing the package
install.packages("cvms")
install.packages("tibble")
```

```
install.packages("vctrs")
install.packages("tidymodels")
install.packages("plotROC")
install.packages("tidymodels")
install.packages("ROCR")
install.packages("caTools")
install.packages("plotROC")
# Loading package
library(caTools)
library(ROCR)
library(tidyverse)
library(readxl)
library(dplyr)
library(cvms)
library(tibble)
library(tidymodels)
library(plotROC)
library(ggplot2)


df <- read_excel("C:\\Users\\Dr.Octopus\\Downloads\\ch datasets\\pre
liminary.xlsx")
df
data<- df[c(1,2,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,22,23,2
4,25,26,27,28,29,30,31)]
data
x <- df$retained
logistic_model <- glm(x ~ .,
data = data,family = "binomial")
logistic_model
summary(logistic_model)
predict_reg <- predict(logistic_mode,data, type = "response")

predict_reg
predict_reg <- ifelse(predict_reg >0.5, 1, 0)
predict_reg


#plot logistic regression curve
ggplot(mtcars, aes(x=x, y=predict_reg)) +
  geom_point(alpha=.5) +
  stat_smooth(method="glm", se=FALSE, method.args = list(family=bino
mial),
              col="red", lty=2)

d_multi <- tibble(x = floor(runif(100) * 3),
predict_reg = floor(runif(100) * 3))

d_multi
conf_mat <- confusion_matrix(targets = d_multi$x,
predictions = d_multi$predict_reg)
Call:  glm(formula = x ~ ., family = "binomial", data = data)

Coefficients:
 (Intercept)           Age        Gender     Ethnicity       Marital
Livewith     Education  palpitations     orthopnea
   -0.761377     -0.005619     -0.019384      0.342394      0.051376
0.434566     -0.011749     -0.083112      0.270789
```

```
   chestpain        nausea         cough       fatigue       dyspnea
edema           PND    tightshoes    weightgain
    0.170977     -0.075170      0.314495     -0.065199     -0.153922
0.322018     -0.054901     -0.114831     -0.096670
         DOE
    0.254748

Degrees of Freedom: 400 Total (i.e. Null);  382 Residual
  (5 observations deleted due to missingness)
Null Deviance:           492.8
Residual Deviance: 458.8    AIC: 496.8
> summary(logistic_model)

Call:
glm(formula = x ~ ., family = "binomial", data = data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0848  -1.1888   0.6624   0.8502   1.4349

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.761377   1.329205  -0.573   0.5668
Age          -0.005619   0.010482  -0.536   0.5919
Gender       -0.019384   0.236557  -0.082   0.9347
Ethnicity     0.342394   0.272325   1.257   0.2086
Marital       0.051376   0.197816   0.260   0.7951
Livewith      0.434566   0.287354   1.512   0.1305
Education    -0.011749   0.084535  -0.139   0.8895
palpitations -0.083112   0.136067  -0.611   0.5413
orthopnea     0.270789   0.126637   2.138   0.0325 *
chestpain     0.170977   0.143793   1.189   0.2344
nausea       -0.075170   0.149471  -0.503   0.6150
cough         0.314495   0.126296   2.490   0.0128 *
fatigue      -0.065199   0.151458  -0.430   0.6669
dyspnea      -0.153922   0.145273  -1.060   0.2894
edema         0.322018   0.140226   2.296   0.0217 *
PND          -0.054901   0.121587  -0.452   0.6516
tightshoes   -0.114831   0.148799  -0.772   0.4403
weightgain   -0.096670   0.123382  -0.784   0.4333
DOE           0.254748   0.132684   1.920   0.0549 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 492.76  on 400  degrees of freedom
Residual deviance: 458.82  on 382  degrees of freedom
  (5 observations deleted due to missingness)
AIC: 496.82

Number of Fisher Scoring iterations: 4

> predict_reg <- predict(logistic_model,
+                        data, type = "response")
>
> predict_reg
```

```
        1         2         3         4         5         6         7
8         9        10        11        12
0.5967331 0.6323268 0.7104618 0.7009473 0.7739618 0.8716476 0.567411
9 0.5626480 0.4659791 0.4659791 0.8973661 0.8991377
       13        14        15        16        17        18        1
9        20        21        22        23        24
0.4524332 0.4572395 0.6640644 0.6597266 0.8466454 0.8466454 0.764076
6 0.7640766 0.5010018 0.5248249 0.6366686 0.6366686
       25        26        27        28        29        30        3
1        32        33        34        35        36
0.4838790 0.4838790 0.5486263 0.5438219 0.6529979 0.7886872 0.760818
4 0.7487531 0.7875330 0.7875330 0.8443756 0.8469057
       37        38        39        40        41        42        4
3        44        45        46        47        48
0.6806645 0.5672638 0.7141384 0.7738298 0.6364144 0.6364144 0.561391
5 0.5661585 0.5841739 0.5794578 0.7254836 0.7293271
       49        50        51        52        53        54        5
5        56        57        58        59        60
0.6107518 0.6882338 0.8389673 0.8415689 0.3808243 0.3808243 0.688885
7 0.6888857 0.2985268 0.2985268 0.4854795 0.4903225
       61        62        63        64        65        66        6
7        68        69        70        71        72
0.8067110 0.8067110 0.7284442 0.7284442 0.5768418 0.8136247 0.576841
8 0.7780285 0.5651109 0.5834862 0.8258655 0.6662786
       73        74        75        76        77        78        7
9        80        81        82        83        84
0.8286355 0.6619548 0.5051244 0.7028075 0.4489660 0.6332553 0.847818
2 0.8568069 0.8478182 0.8321881 0.7826672 0.6483693
       85        86        87        88        89        90        9
1        92        93        94        95        96
0.7826672 0.6439375 0.7583106 0.6853312 0.7618453 0.6429108 0.661482
8 0.6047216 0.5914334 0.5961088 0.6950988 0.7623085
       97        98        99       100       101       102       10
3       104       105       106       107       108
0.6329953 0.5013106 0.7623085 0.3857503 0.7282173 0.6716896 0.750569
2 0.6903614 0.5680919 0.6093859 0.7188684 0.6689818
      109       110       111       112       113       114       11
5       116       117       118       119       120
0.7685953 0.4554078 0.7651299 0.3894427 0.7490328 0.6898350 0.752659
0 0.9420743 0.6856725 0.8913044 0.6953808 0.7108841
      121       122       123       124       125       126       12
7       128       129       130       131       132
0.8494702 0.8269445 0.8047037 0.6738341 0.8411247 0.6738341 0.828808
6 0.8999281 0.8288086 0.8999281 0.5782272 0.9211502
      133       134       135       136       137       138       13
9       140       141       142       143       144
0.5002512 0.9211502 0.7846368 0.7846368 0.7858227 0.9097901 0.761103
2 0.9113683 0.6539775 0.7546600 0.6495782 0.7011612
      145       146       147       148       149       150       15
1       152       153       154       155       156
0.7679719 0.8326029 0.7645000 0.8049601 0.8456468 0.6834212 0.804960
1 0.6792126 0.5510129 0.5462128 0.6738077 0.8925150
      157       158       159       160       161       162       16
3       164       165       166       167       168
0.4922996 0.8943604 0.8069927 0.7840165 0.7882669 0.7995977 0.713938
1 0.7250176 0.5974025        NA 0.5927318        NA
      169       170       171       172       173       174       17
5       176       177       178       179       180
```

0.5066844 0.6050000 0.4814869 0.9449578 0.6003585 0.8684660 0.938757
5 0.8854341 0.6541687 0.6881182 0.6906384 0.7496797
      181   182   183   184   185   186   18
7   188   189   190   191   192
0.5852863 0.7496797 0.6600279 0.8861840 0.5067258 0.8842143 0.712975
4 0.7225811 0.7358891 0.4182419 0.7186788 0.8106811
      193   194   195   196   197   198   19
9   200   201   202   203   204
0.4527351 0.8794508 0.8106811 0.7829718 0.8812917 0.8153546 0.870743
7 0.7709782 0.8128259 0.8157570 0.7772901 0.7181794
      205   206   207   208   209   210   21
1   212   213   214   215   216
0.7413077 0.7541989 0.7569240 0.7211466 0.8209692 0.8141278 0.773436
1 0.8268247 0.8170432 0.8240316 0.8537906 0.7758412
      217   218   219   220   221   222   22
3   224   225   226   227   228
0.8260034 0.7190334 0.7018554 0.7741780 0.6977835 0.7932039 0.857890
4 0.6847153 0.4773191 0.7563261 0.8178562 0.7160522
      229   230   231   232   233   234   23
5   236   237   238   239   240
0.5587995 0.5194803 0.7306429 0.7781698 0.8030150 0.7141058 0.841290
5 0.8438615 0.5430032 0.5430032 0.6160561 0.6160561
      241   242   243   244   245   246   24
7   248   249   250   251   252
0.6177097 0.6131221 0.8003051 0.7971893 0.7103621 0.7870854 0.836100
9 0.8109496 0.4318725 0.4318725 0.5939829 0.5399946
      253   254   255   256   257   258   25
9   260   261   262   263   264
0.7459838 0.6290144 0.8526739 0.4640994 0.3736685 0.4839590 0.697271
5 0.7334753 0.7758565 0.5731068 0.6172947 0.8734599
      265   266   267   268   269   270   27
1   272   273   274   275   276
0.9319992 0.6563739 0.7170660 0.5909793 0.7292835 0.3571785 0.715554
2 0.6065810 0.6488806 0.6905453 0.6318523 0.6799242
      277   278   279   280   281   282   28
3   284   285   286   287   288
0.2522425 0.6914316 0.7634325 0.6996072 0.6860196 0.6320556 0.642558
3 0.7538938 0.7644229 0.5925119 0.4729861 0.8539949
      289   290   291   292   293   294   29
5   296   297   298   299   300
0.8467740 0.7383468 0.5646174 0.7577874 0.7386519 0.7219253 0.900653
1 0.6599794 0.6822318 0.5579826 0.7608534 0.6375380
      301   302   303   304   305   306   30
7   308   309   310   311   312
0.6930088 0.4839174 0.7614889 0.8112783 0.7299990 0.8145881 0.869788
0 0.5306827 0.8487983 0.4332397 0.8595331 0.6873276
      313   314   315   316   317   318   31
9   320   321   322   323   324
0.5469478 0.6464302 0.8427496 0.7123521 0.6970093 0.8942847 0.701722
5 0.7431818 0.6293965 0.5916778 0.7048411 0.6984685
      325   326   327   328   329   330   33
1   332   333   334   335   336
0.5208604 0.8397614 0.6980302 0.4496326 0.7344777 0.8923843 0.568269
9 0.8592555 0.5081029 0.8463528 0.6111503 0.5059088
      337   338   339   340   341   342   34
3   344   345   346   347   348
0.6625890 0.8739942 0.6913611 0.3870435 0.7518424 0.4965655 0.815978
3 0.7245696 0.4966917 0.8673229 0.7994543 0.6917184

```
      349       350       351       352       353       354       35
5       356       357       358       359       360
0.8689242 0.7053013 0.5815129 0.7936684 0.4827444 0.3900614 0.603123
5 0.6461643 0.7361427 0.6736795 0.8711033 0.7157650
      361       362       363       364       365       366       36
7       368       369       370       371       372
0.4572729 0.6513582 0.5599371 0.8468324 0.6966746 0.7269729 0.581457
7 0.7477120 0.9349720 0.8342192 0.5028786 0.6018066
      373       374       375       376       377       378       37
9       380       381       382       383       384
0.8560305 0.7258270 0.7443381 0.7910161 0.5247357 0.5992575 0.679476
2 0.7051483 0.7471050 0.7291205 0.7769276 0.6937445
      385       386       387       388       389       390       39
1       392       393       394       395       396
0.7123533 0.6140431 0.6953792 0.6098041 0.6202082 0.6631964 0.895107
0 0.6021557 0.5447013 0.6236694 0.7553344 0.6853452
      397       398       399       400       401       402       40
3       404       405       406
0.8654841 0.6888848 0.5192177 0.7563261 0.7409323 0.7120946 0.554015
3 0.5194803 0.7306429 0.7748058
```

(2) Heart Health Data

First of all, I made a copy of the heart health data to generate a separate data frame in order to use a logistic model to forecast if a person will seek medical attention in two days or less. I then made a new column called "delay_day_2" with values of 1 if the value in delaydays is less than or equal to 2, else 0, and 0 otherwise. I removed all the pointless columns (ID, delaydays) from the data frame before fitting the logistic regression model to the dataset.

I then used the prepared data to fit the logistic model by designating the delay_day_2 column as the dependent variable and the other factors as independent variables. This is a summary of the fitted logistic model:

```
Call:  glm(formula = x ~ ., family = "binomial", data = data)

Coefficients:
 (Intercept)           Age        Gender     Ethnicity       Marital
Livewith     Education  palpitations     orthopnea
   -0.761377     -0.005619     -0.019384      0.342394      0.051376
0.434566     -0.011749     -0.083112      0.270789
   chestpain        nausea         cough       fatigue       dyspnea
edema           PND    tightshoes    weightgain
    0.170977     -0.075170      0.314495     -0.065199     -0.153922
0.322018     -0.054901     -0.114831     -0.096670
         DOE
    0.254748

Degrees of Freedom: 400 Total (i.e. Null);  382 Residual
  (5 observations deleted due to missingness)
Null Deviance:           492.8
Residual Deviance: 458.8    AIC: 496.8
> summary(logistic_model)

Call:
glm(formula = x ~ ., family = "binomial", data = data)
```

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0848  -1.1888   0.6624   0.8502   1.4349

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.761377   1.329205  -0.573   0.5668
Age          -0.005619   0.010482  -0.536   0.5919
Gender       -0.019384   0.236557  -0.082   0.9347
Ethnicity     0.342394   0.272325   1.257   0.2086
Marital       0.051376   0.197816   0.260   0.7951
Livewith      0.434566   0.287354   1.512   0.1305
Education    -0.011749   0.084535  -0.139   0.8895
palpitations -0.083112   0.136067  -0.611   0.5413
orthopnea     0.270789   0.126637   2.138   0.0325 *
chestpain     0.170977   0.143793   1.189   0.2344
nausea       -0.075170   0.149471  -0.503   0.6150
cough         0.314495   0.126296   2.490   0.0128 *
fatigue      -0.065199   0.151458  -0.430   0.6669
dyspnea      -0.153922   0.145273  -1.060   0.2894
edema         0.322018   0.140226   2.296   0.0217 *
PND          -0.054901   0.121587  -0.452   0.6516
tightshoes   -0.114831   0.148799  -0.772   0.4403
weightgain   -0.096670   0.123382  -0.784   0.4333
DOE           0.254748   0.132684   1.920   0.0549 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 492.76  on 400  degrees of freedom
Residual deviance: 458.82  on 382  degrees of freedom
  (5 observations deleted due to missingness)
AIC: 496.82
```

A logistic model to determine whether a patient will seek medical attention on or before the cohort's average number of delay days.

Once more, I made a copy of the heart health data and generated a different data frame. I then established a new column called "delay day avg," with values of 1 if the value in the "delaydays" column is less than or equal to its mean value and 0 otherwise. I removed all the pointless columns (ID, delaydays) from the data frame before fitting the logistic regression model to the dataset.

I then used the prepared data to fit the logistic model by designating the delay day avg column as the dependent variable and the other factors as independent variables. This is a summary of the fitted logistic model:

Code:

```
# Installing the package

install.packages("cvms")
```

```
install.packages("tibble")
install.packages("vctrs")
install.packages("tidymodels")
install.packages("plotROC")
install.packages("tidymodels")
install.packages("ROCR")
install.packages("caTools")
install.packages("plotROC")
# Loading package
library(caTools)
library(ROCR)
library(tidyverse)
library(readxl)
library(dplyr)
library(cvms)
library(tibble)
library(tidymodels)
library(plotROC)
library(ggplot2)
# load the dataset
df <- read_excel("C:\\Users\\Dr.Octopus\\Downloads\\ch datasets\\heart-health-data.xls")
df
df <- subset(df, select = -c(ID))
df
data<- df[c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18)]
data
df$delaydays
df$delaydays <- as.factor(ifelse(df$delaydays > 1, 1, 0))
x<-df$delaydays
x
#train_test
```

```
#df$delaydays
# Training model
logistic_model <- glm(x ~ .,
                      data = data,
                      family = "binomial")
logistic_model
summary(logistic_model)
predict_reg <- predict(logistic_model,
                       data, type = "response")

predict_reg
predict_reg <- ifelse(predict_reg >0.5, 1, 0)
predict_reg

basic_table <- table(x,predict_reg)
print(basic_table)
cfm <- as_tibble(basic_table)
cfm
################################
df$delaydays <- as.factor(ifelse(df$delaydays > 1, 1, 0))
x<-df$delaydays
x
#train_test
#df$delaydays
# Training model
logistic_model <- glm(x ~ .,
                      data = data,
                      family = "binomial")
logistic_model1
summary(logistic_model)
predict_reg <- predict(logistic_model,
```

```
                    data, type = "response")


predict_reg
predict_reg <- ifelse(predict_reg >0.5, 1, 0)
predict_reg


basic_table <- table(x,predict_reg)
print(basic_table)
cfm <- as_tibble(basic_table)
cfm
```