20BDS0146

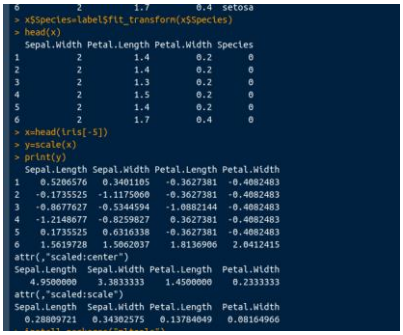
VENNELA G

PROGRAMMING FOR DATA SCIENCE

LAB ASSESSMENT 3

1. Perform Label encoding on IRIS Dataset

```
#label encoding
x=iris[-c(1,33)]
sum(is.na(x))
colSums(is.na(x))
install.packages("superml")
library(superml)
label=LabelEncoder$new()
x$Species=label$fit_transform(x$Species)
head(x)
```

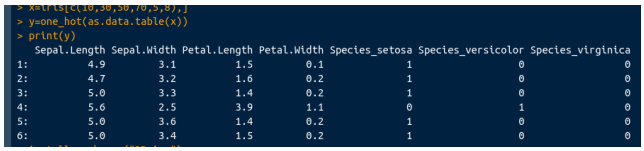


```
6           2          1.7         0.4  setosa
> x$Species=label$flt_transform(x$Species)
> head(x)
  Sepal.Width Petal.Length Petal.Width Species
1           2          1.4         0.2       0
2           2          1.4         0.2       0
3           2          1.3         0.2       0
4           2          1.5         0.2       0
5           2          1.4         0.2       0
6           2          1.7         0.4       0
> x=head(iris[-5])
> y=scale(x)
> print(y)
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1    0.5206576   0.3401105   -0.3627381  -0.4082483
2   -0.1735525  -1.1175060   -0.3627381  -0.4082483
3   -0.8677627  -0.5344594   -1.0882144  -0.4082483
4   -1.2148677  -0.8259827    0.3627381  -0.4082483
5    0.1735525   0.6316338   -0.3627381  -0.4082483
6    1.5619728   1.5062037    1.8136906   2.0412415
attr(,"scaled:center")
Sepal.Length  Sepal.Width Petal.Length  Petal.Width
   4.9500000    3.3833333    1.4500000    0.2333333
attr(,"scaled:scale")
Sepal.Length  Sepal.Width Petal.Length  Petal.Width
  0.28809721   0.34302575   0.13784049   0.08164966
```

2. Perform One-hot encoding on IRIS Dataset

```
#one hot encoding

install.packages("mltools")
library(mltools)
install.packages("data.table")
library(data.table)
x=iris[c(10,30,50,70,5,8),]
y=one_hot(as.data.table(x))
print(y)
```



```
> x=iris[c(10,30,50,70,5,8),]
> y=one_hot(as.data.table(x))
> print(y)
   Sepal.Length Sepal.Width Petal.Length Petal.Width Species_setosa Species_versicolor Species_virginica
1:          4.9         3.1          1.5         0.1              1                 0                0
2:          4.7         3.2          1.6         0.2              1                 0                0
3:          5.0         3.3          1.4         0.2              1                 0                0
4:          5.6         2.5          3.9         1.1              0                 1                0
5:          5.0         3.6          1.4         0.2              1                 0                0
6:          5.0         3.4          1.5         0.2              1                 0                0
```

3.Feature scaling or standardization

a. Normalization

#normalization

install.packages("BBmisc")
library(BBmisc)
y=normalize(x,method="standardize")
print(y)

```
> y=normalize(x,method="standardize")
> print(y)
   Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
10   -0.4428074 -0.22140372   -0.3868963  -0.6177539     setosa
30   -1.1070186  0.04428074   -0.2859668  -0.3530023     setosa
50   -0.1107019  0.30996521   -0.4878258  -0.3530023     setosa
70    1.8819316 -1.81551052    2.0354109   2.0297630 versicolor
5    -0.1107019  1.10701861   -0.4878258  -0.3530023     setosa
8    -0.1107019  0.57564968   -0.3868963  -0.3530023     setosa
```

b.Z-scale
#Z-scale
names(x)
x=iris[-5]
z=sapply(x,function(x)((x-mean(x))/sd(x)))
print(head(z))

```
> print(head(z))
     Sepal.Length Sepal.Width Petal.Length Petal.Width
[1,]   -0.8976739  1.01560199    -1.335752   -1.311052
[2,]   -1.1392005 -0.13153881    -1.335752   -1.311052
[3,]   -1.3807271  0.32731751    -1.392399   -1.311052
[4,]   -1.5014904  0.09788935    -1.279104   -1.311052
[5,]   -1.0184372  1.24503015    -1.335752   -1.311052
[6,]   -0.5353840  1.93331463    -1.165809   -1.048667
```

4.Find the principal components of IRIS dataset
#principal components of iris data set

x=iris[-5]
my_pca=prcomp(x, scale = TRUE,center = TRUE, retx = T)
names(iris)
summary(iris)
my_pca$rotation
my_pca$sdev

```
> names(iris)
[1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"  "Species"
> summary(iris)
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width          Species
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100   setosa    :50
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300   versicolor:50
 Median :5.800   Median :3.000   Median :4.350   Median :1.300   virginica :50
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
> my_pca$rotation
                    PC1         PC2        PC3        PC4
Sepal.Length  0.5210659 -0.37741762  0.7195664  0.2612863
Sepal.Width  -0.2693474 -0.92329566 -0.2443818 -0.1235096
Petal.Length  0.5804131 -0.02449161 -0.1421264 -0.8014492
Petal.Width   0.5648565 -0.06694199 -0.6342727  0.5235971
> my_pca$sdev
[1] 1.7083611 0.9560494 0.3830886 0.1439265
```

5.House rent prediction using linear regression

#house rent prediction

#Read Dataset
install.packages("mlbench")
library(mlbench)
data(BostonHousing)
x=BostonHousing
str(x)
names(x)

#Check NA in dataset
sum(is.na(x))

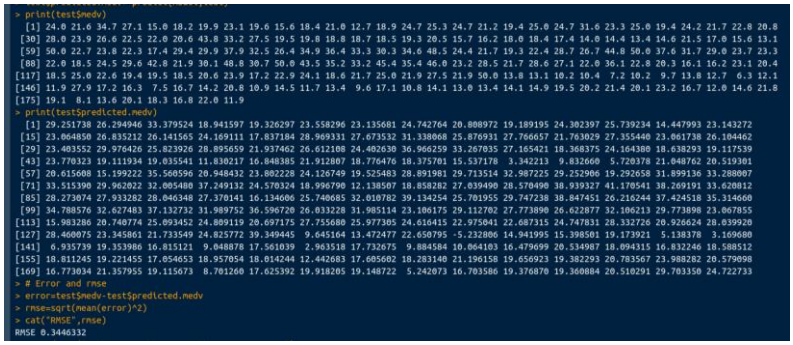
```
> x=BostonHousing
> str(x)
'data.frame':   506 obs. of  14 variables:
 $ crim   : num  0.00632 0.02731 0.02729 0.03237 0.06905 ...
 $ zn     : num  18 0 0 0 0 0 12.5 12.5 12.5 12.5 ...
 $ indus  : num  2.31 7.07 7.07 2.18 2.18 2.18 7.87 7.87 7.87 7.87 ...
 $ chas   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ nox    : num  0.538 0.469 0.469 0.458 0.458 0.458 0.524 0.524 0.524 0.524 ...
 $ rm     : num  6.58 6.42 7.18 7 7.15 ...
 $ age    : num  65.2 78.9 61.1 45.8 54.2 58.7 66.6 96.1 100 85.9 ...
 $ dis    : num  4.09 4.97 4.97 6.06 6.06 ...
 $ rad    : num  1 2 2 3 3 3 5 5 5 5 ...
 $ tax    : num  296 242 242 222 222 222 311 311 311 311 ...
 $ ptratio: num  15.3 17.8 17.8 18.7 18.7 18.7 15.2 15.2 15.2 15.2 ...
 $ b      : num  397 397 393 395 397 ...
 $ lstat  : num  4.98 9.14 4.03 2.94 5.33 ...
 $ medv   : num  24 21.6 34.7 33.4 36.2 28.7 22.9 27.1 16.5 18.9 ...
> names(x)
 [1] "crim"    "zn"      "indus"   "chas"    "nox"     "rm"      "age"     "dis"     "rad"     "tax"     "ptratio" "b"       "lstat"   "medv"
> #Check NA in dataset
> sum(is.na(x))
 [1] 0
```

#split dataset
install.packages('caTools')
library(caTools)
split=sample.split(x,SplitRatio =0.7)
train=subset(x,split==TRUE)
test=subset(x,split==FALSE)

#The General Linear regression model in R :
#Univariate Model : model<−lm(y∼x,data)
Multivariate Model : model<−lm(y∼.,data)
#medv is the target variable, predicted using crim,rm,tax,lstat
model=lm(medv ~ crim + rm + tax + lstat , data = train)
summary(model)

#Prediction
test$predicted.medv= predict(model,test)
print(test$medv)
print(test$predicted.medv)

# Error and rmse
error=test$medv-test$predicted.medv
rmse=sqrt(mean(error)^2)
cat("RMSE",rmse)



6. Medical diagnosis for disease spread pattern Using SVM

#medical diagnosis

x=read.csv("C:\\Vennela\\Downloads\\Cancer_Data.csv")
names(x)

```
> names(x)
 [1] "id"                      "diagnosis"               "radius_mean"
 [4] "texture_mean"            "perimeter_mean"          "area_mean"
 [7] "smoothness_mean"         "compactness_mean"        "concavity_mean"
[10] "concave.points_mean"     "symmetry_mean"           "fractal_dimension_mean"
[13] "radius_se"               "texture_se"              "perimeter_se"
[16] "area_se"                 "smoothness_se"           "compactness_se"
[19] "concavity_se"            "concave.points_se"       "symmetry_se"
[22] "fractal_dimension_se"    "radius_worst"            "texture_worst"
[25] "perimeter_worst"         "area_worst"              "smoothness_worst"
[28] "compactness_worst"       "concavity_worst"         "concave.points_worst"
[31] "symmetry_worst"          "fractal_dimension_worst" "X"
> x=x[-c(1,33)]
> sum(is.na(x))
[1] 0
> colSums(is.na(x))
             diagnosis            radius_mean           texture_mean
                     0                      0                      0
        perimeter_mean              area_mean        smoothness_mean
                     0                      0                      0
       compactness_mean         concavity_mean     concave.points_mean
                     0                      0                      0
         symmetry_mean fractal_dimension_mean              radius_se
                     0                      0                      0
            texture_se          perimeter_se               area_se
                     0                      0                      0
         smoothness_se         compactness_se           concavity_se
                     0                      0                      0
      concave.points_se           symmetry_se    fractal_dimension_se
                     0                      0                      0
          radius_worst         texture_worst        perimeter_worst
                     0                      0                      0
            area_worst       smoothness_worst       compactness_worst
                     0                      0                      0
        concavity_worst    concave.points_worst          symmetry_worst
                     0                      0                      0
fractal_dimension_worst
                     0
```

x=x[-c(1,33)]

#Check NA in dataset
sum(is.na(x))
colSums(is.na(x))

#Label Encoder
library(superml)
label=LabelEncoder$new()
x$diagnosis=label$fit_transform(x$diagnosis)
head(x)

```
> head(x)
  diagnosis radius_mean texture_mean perimeter_mean area_mean smoothness_mean
1         0       17.99        10.38         122.80    1001.0         0.11840
2         0       20.57        17.77         132.90    1326.0         0.08474
3         0       19.69        21.25         130.00    1203.0         0.10960
4         0       11.42        20.38          77.58     386.1         0.14250
5         0       20.29        14.34         135.10    1297.0         0.10030
6         0       12.45        15.70          82.57     477.1         0.12780
  compactness_mean concavity_mean concave.points_mean symmetry_mean fractal_dimension_mean
1          0.27760         0.3001             0.14710        0.2419                0.07871
2          0.07864         0.0869             0.07017        0.1812                0.05667
3          0.15990         0.1974             0.12790        0.2069                0.05999
4          0.28390         0.2414             0.10520        0.2597                0.09744
5          0.13280         0.1980             0.10430        0.1809                0.05883
6          0.17000         0.1578             0.08089        0.2087                0.07613
  radius_se texture_se perimeter_se area_se smoothness_se compactness_se concavity_se
1    1.0950     0.9053        8.589  153.40      0.006399        0.04904      0.05373
2    0.5435     0.7339        3.398   74.08      0.005225        0.01308      0.01860
3    0.7456     0.7869        4.585   94.03      0.006150        0.04006      0.03832
4    0.4956     1.1560        3.445   27.23      0.009110        0.07458      0.05661
5    0.7572     0.7813        5.438   94.44      0.011490        0.02461      0.05688
6    0.3345     0.8902        2.217   27.19      0.007510        0.03345      0.03672
  concave.points_se symmetry_se fractal_dimension_se radius_worst texture_worst
1           0.01587     0.03003             0.006193        25.38         17.33
2           0.01340     0.01389             0.003532        24.99         23.41
3           0.02058     0.02250             0.004571        23.57         25.53
4           0.01867     0.05963             0.009208        14.91         26.50
5           0.01885     0.01756             0.005115        22.54         16.67
6           0.01137     0.02165             0.005082        15.47         23.75
  perimeter_worst area_worst smoothness_worst compactness_worst concavity_worst
1          184.60     2019.0            0.1622            0.6656          0.7119
2          158.80     1956.0            0.1238            0.1866          0.2416
3          152.50     1709.0            0.1444            0.4245          0.4504
4           98.87      567.7            0.2098            0.8663          0.6869
5          152.20     1575.0            0.1374            0.2050          0.4000
6          103.40      741.6            0.1791            0.5249          0.5355
  concave.points_worst symmetry_worst fractal_dimension_worst
1               0.2654          0.4601                0.11890
2               0.1860          0.2750                0.08902
3               0.2430          0.3613                0.08758
4               0.2575          0.6638                0.17300
5               0.1625          0.2364                0.07678
6               0.1741          0.3985                0.12440
```

#Train-test split
library(caTools)
split=sample.split(x$diagnosis,SplitRatio =0.7)
train=subset(x,split==TRUE)
test=subset(x,split==FALSE)

#SVM
install.packages('e1071')
library(e1071)
train[-1]=scale(train[-1])
test[-1]=scale(test[-1])
names(train)

```
classifier = svm(formula = diagnosis ~ .,
                 data = train,
                 type = 'C-classification',
                 kernel = 'linear')
```

#Prediction

Diag_pred = predict(classifier, newdata = test[-1])

# Making the Confusion Matrix
cm = table(test[,1], Diag_pred)
print(cm)

```
> names(train)
 [1] "diagnosis"               "radius_mean"             "texture_mean"
 [4] "perimeter_mean"          "area_mean"               "smoothness_mean"
 [7] "compactness_mean"        "concavity_mean"          "concave.points_mean"
[10] "symmetry_mean"           "fractal_dimension_mean"  "radius_se"
[13] "texture_se"              "perimeter_se"            "area_se"
[16] "smoothness_se"           "compactness_se"          "concavity_se"
[19] "concave.points_se"       "symmetry_se"             "fractal_dimension_se"
[22] "radius_worst"            "texture_worst"           "perimeter_worst"
[25] "area_worst"              "smoothness_worst"        "compactness_worst"
[28] "concavity_worst"         "concave.points_worst"    "symmetry_worst"
[31] "fractal_dimension_worst"
> classifier = svm(formula = diagnosis ~ ., data = train, type = 'C-classification', kernel = '
near')
> Diag_pred = predict(classifier, newdata = test[-1])
> cm = table(test[,1], Diag_pred)
> print(cm)
   Diag_pred
      0   1
  0  58   6
  1   2 105
> |
```