**3**

## FACULTY OF ENGINEERING AND TECHNOLOGY

## DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

**ENCS3130 Linux Laboratory**

**Project #1**

**Shell Scripting Project**

………….………………………………………………………………………………………………

**Student Name:** Osaid Hamza **ID .No:** 1200875

**Section No. :** 2M

**Instructor:** Dr. Mohammad Jubran

**Date:** 1/ 1/2022

# Abstract:

The aims of this project to encode and scale features of a dataset written in a text file. And a features could be categorical or numeric. For categorical features, there are two types of encoding: label encoding and one-hot encoding.

In one-hot encoding, each category is mapped to a vector that contains 1 and 0 denoting the presence or absence of the feature. The number of vectors depends on the number of categories for features.

In label encoding, each category is assigned a value from 0 through N, where N is the number of categories for the features. There is no relation or order between these assignments.

For two kinds of features, there is a MinMax scaling technique that makes a new values for the chosen feature becomes between 0 and 1.

Contents:

Procedure 1…………………………………………………………………………………………………….

Code…………………………………………………………………………………………………

Description………………………………………………………………………………………

Output………………………………………………………………………………………………………………..

Procedure 2…………………………………………………………………………………………………….

Code…………………………………………………………………………………………………

Description………………………………………………………………………………………

Output………………………………………………………………………………………………………………..

Procedure 3…………………………………………………………………………………………………….

Code…………………………………………………………………………………………………

Description………………………………………………………………………………………

Output………………………………………………………………………………………………………………..

Procedure 4…………………………………………………………………………………………………….

Code…………………………………………………………………………………………………

Description………………………………………………………………………………………

Output………………………………………………………………………………………………………………..

Procedure 5…………………………………………………………………………………………………….

Code…………………………………………………………………………………………………

Description………………………………………………………………………………………

Output………………………………………………………………………………………………………………..

Procedure 6…………………………………………………………………………………………………….

Code…………………………………………………………………………………………………

Description………………………………………………………………………………………

Output………………………………………………………………………………………………………………..

Procedure 7…………………………………………………………………………………………………….

Code…………………………………………………………………………………………………

Description………………………………………………………………………………………

Output………………………………………………………………………………………………………………..

Procedure 8…………………………………………………………………………………………………….

Code…………………………………………………………………………………………………

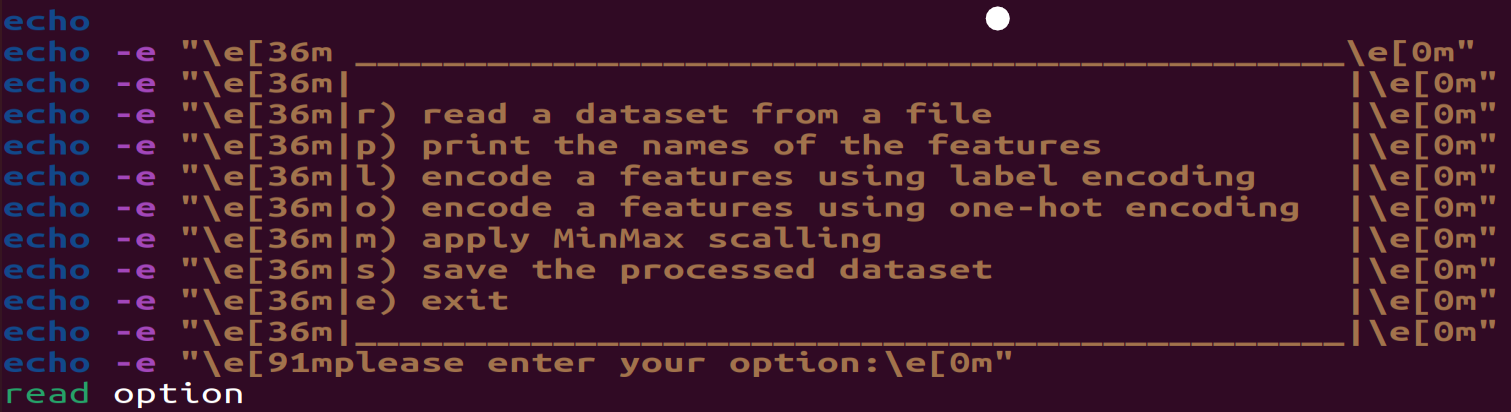
Description………………………………………………………………………………………

Output………………………………………………………………………………………………………………..

Full program code…………………………………………………………………………………………………….

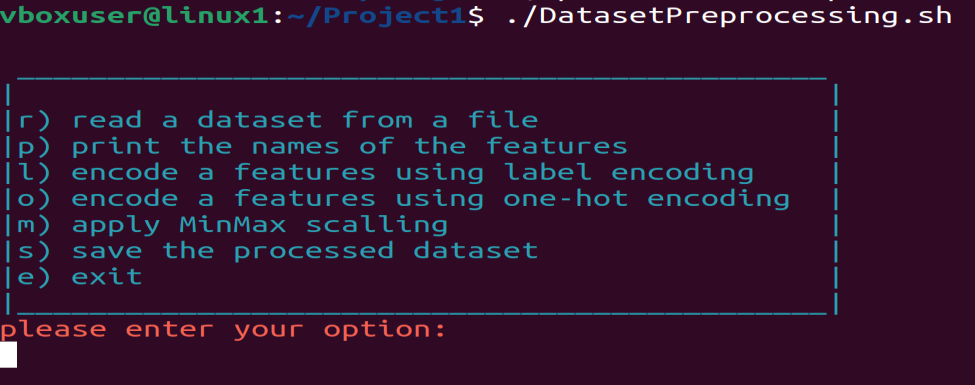
Procedure 1:

Code:



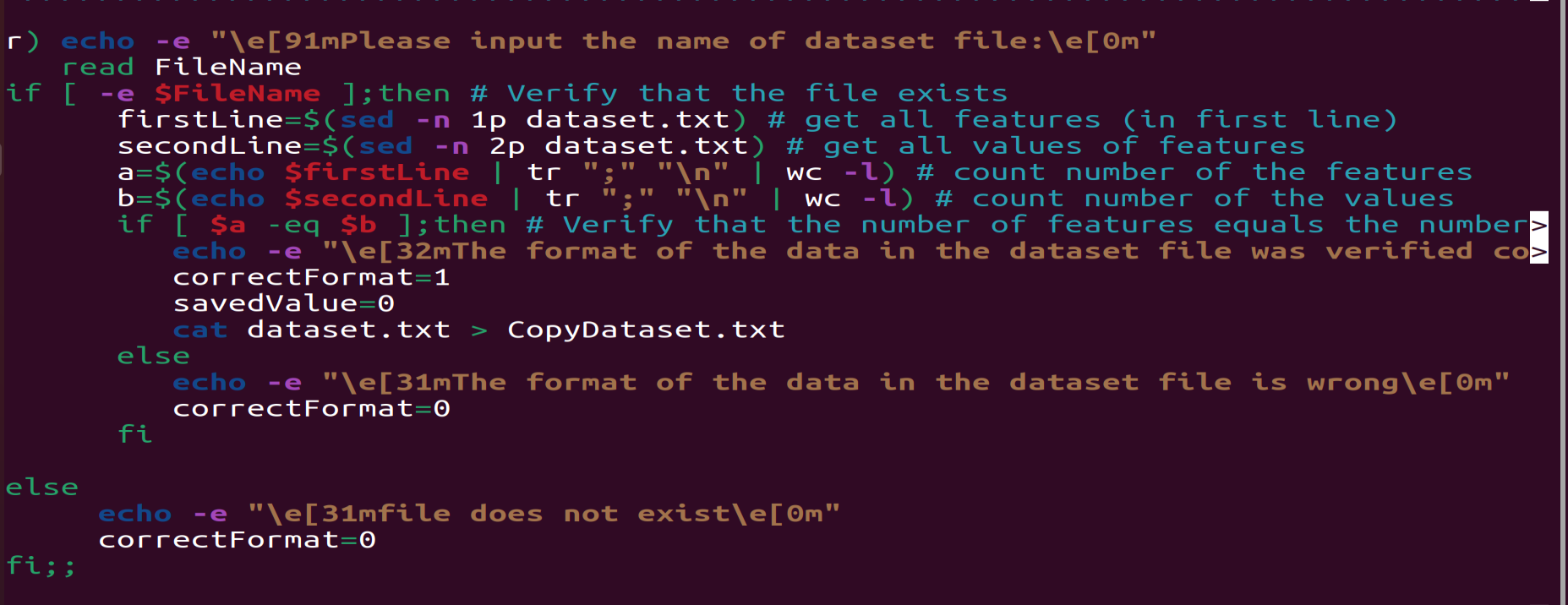
Description: when the program starting, it will print on the screen the main menu and ask the user to select an option.

Output:



Procedure 2, when the user enters option “r”:

Code:

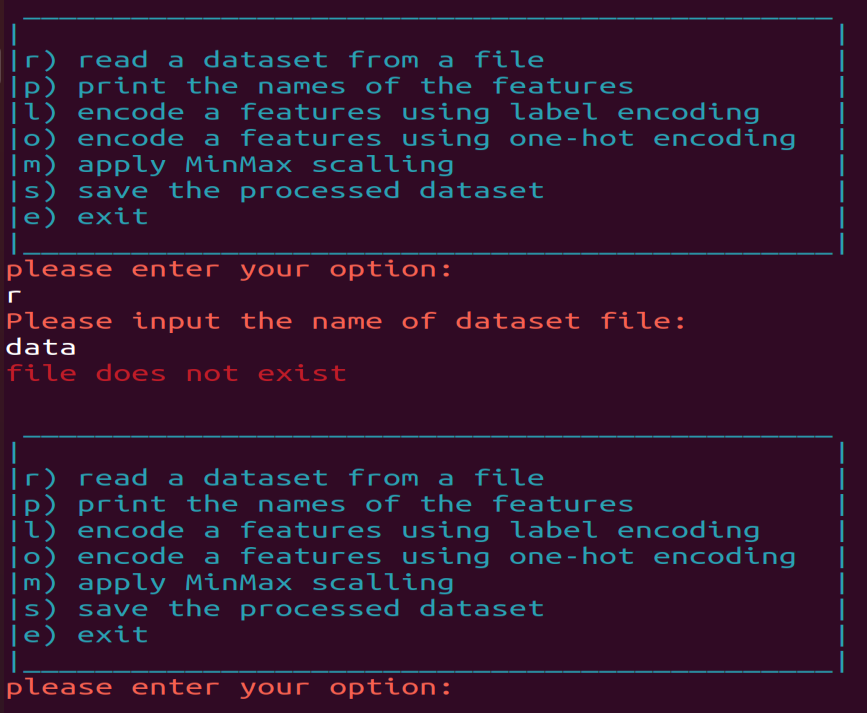


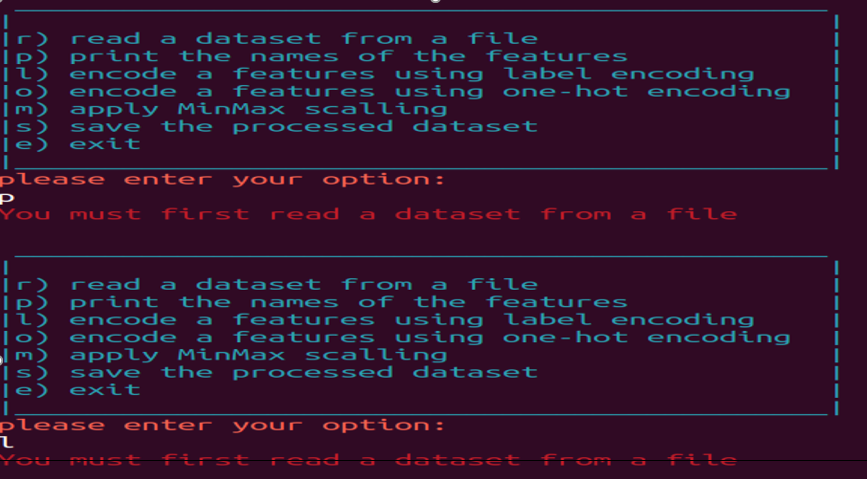
Description:

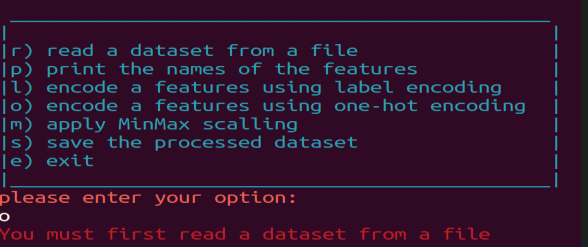
The program must check if the file exists, and while there, it must check that the file format is correct. Otherwise, it should print “file doesn’t exist” to the user if it doesn’t exist, or print “the format of the data in the dataset file is wrong” if the format is wrong.

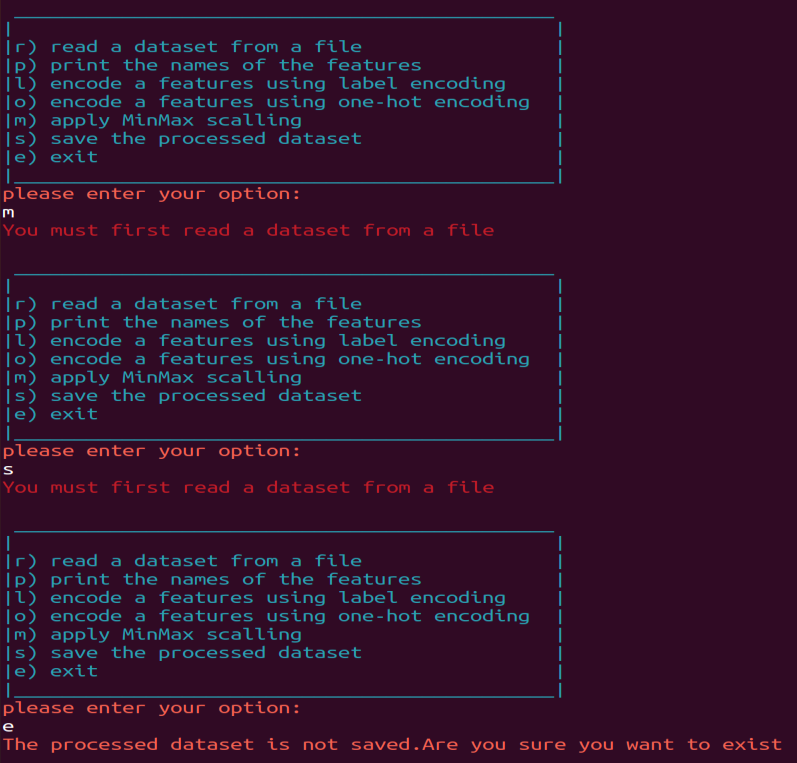
And while a wrong occurs ) whether exist or format), the user can’t choose any other option except “e”. Otherwise (the user entered other option) it should print “You must first read a dataset from a file” .

Output: when the user entered a file doesn’t exisit,then he can’t choose any option except option “e”:



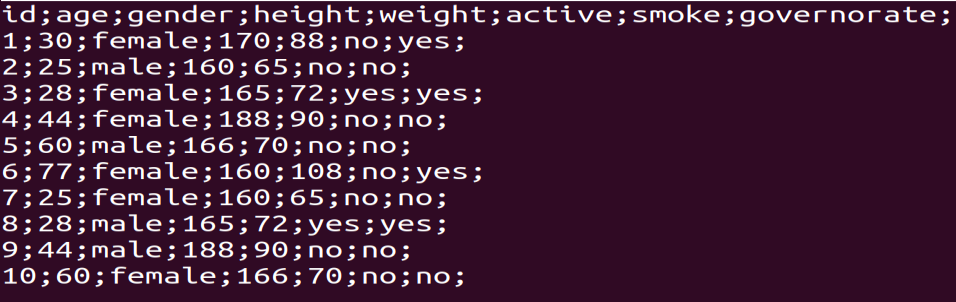
****

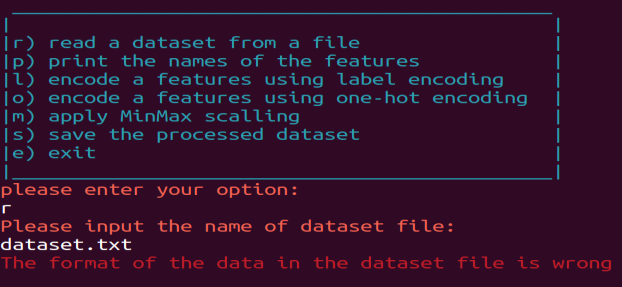


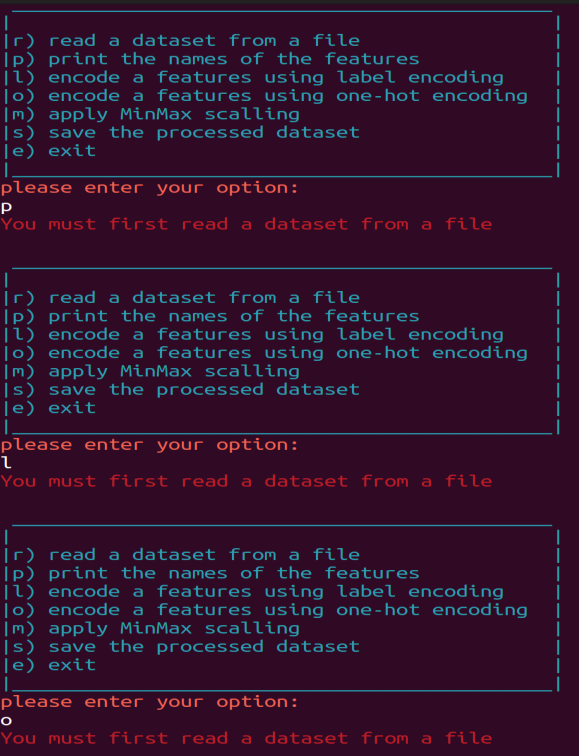


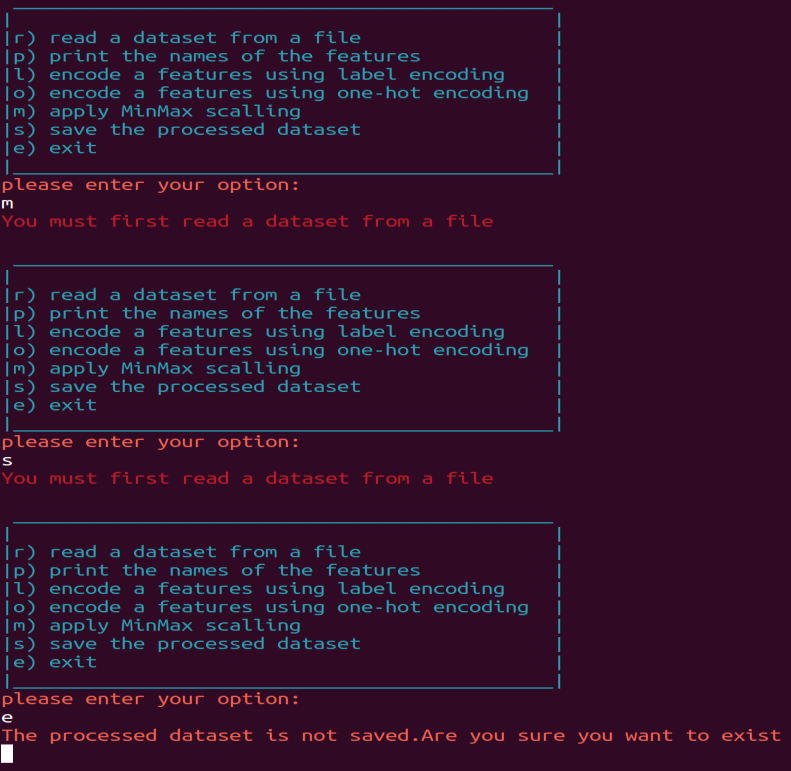
- When the user has entered an existing file but the file format is wrong then he can’t choose any option except option “e”:

The file:



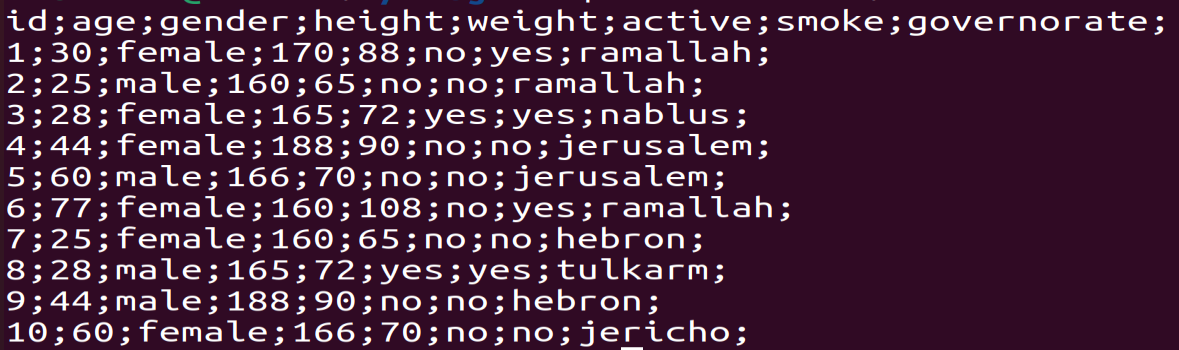


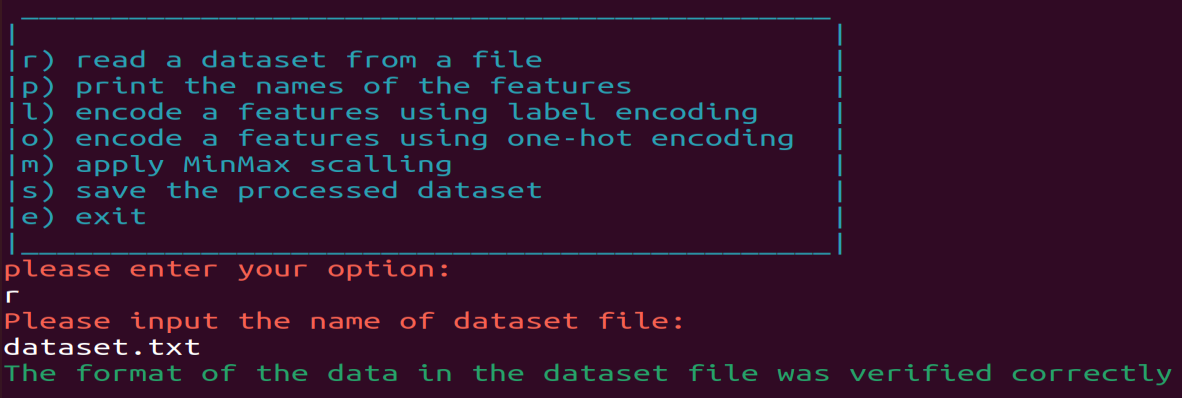




- When the user has entered an existing file and the file format is correct.

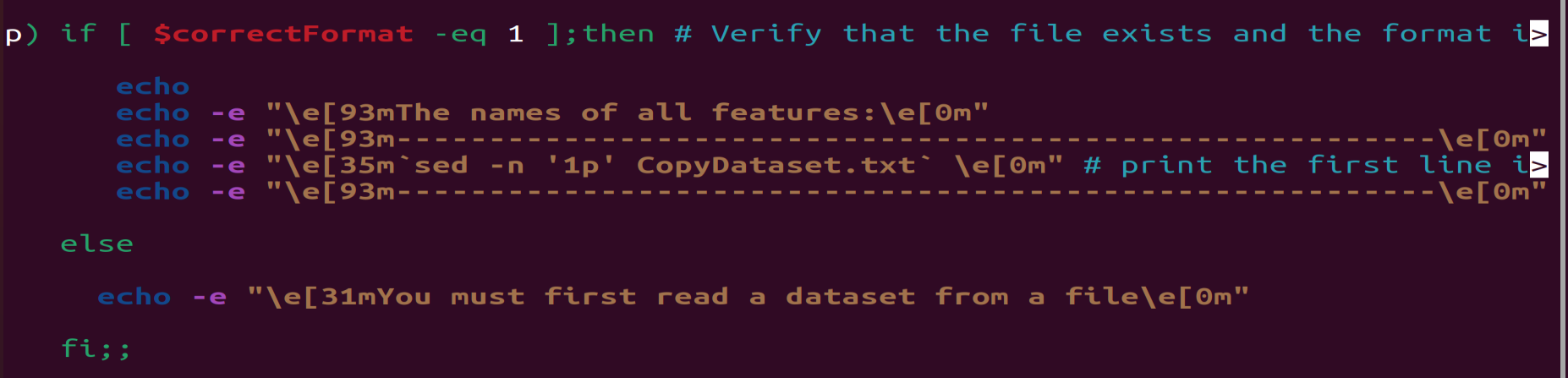
The file:





Procedure 3, when the user enters option “p”:

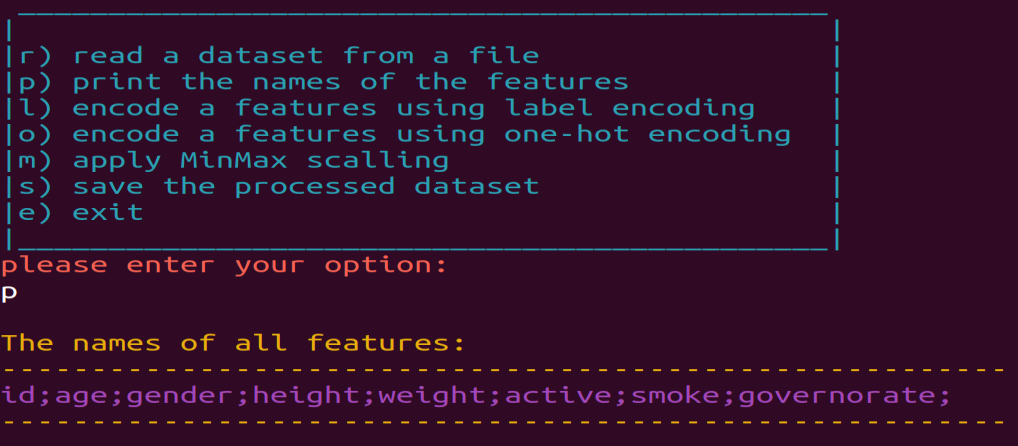
Code:



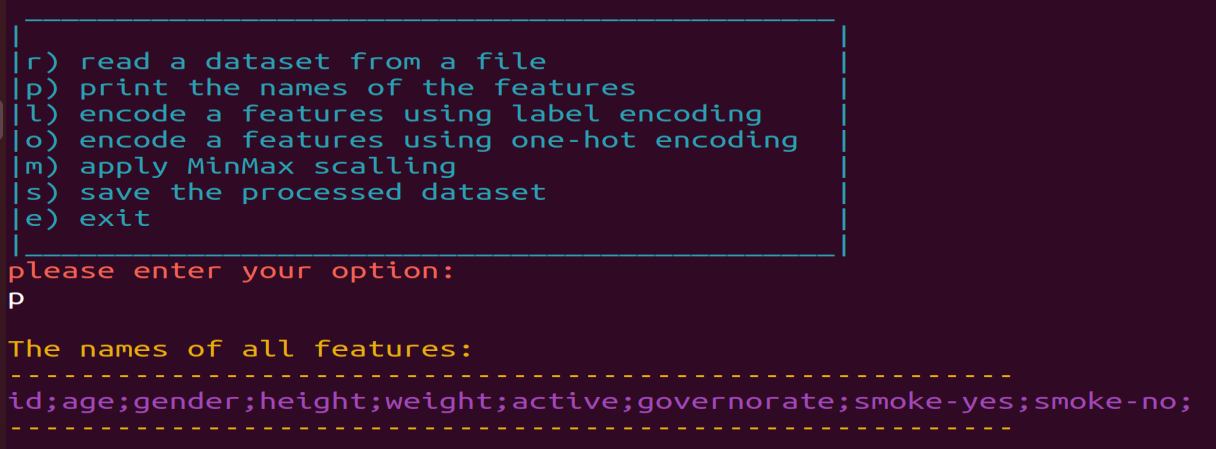
Description:

The program will print the names of all features of the dataset on the screen, by getting the first line in dataset file using “ sed –n ‘1p’ filename ”

Output:

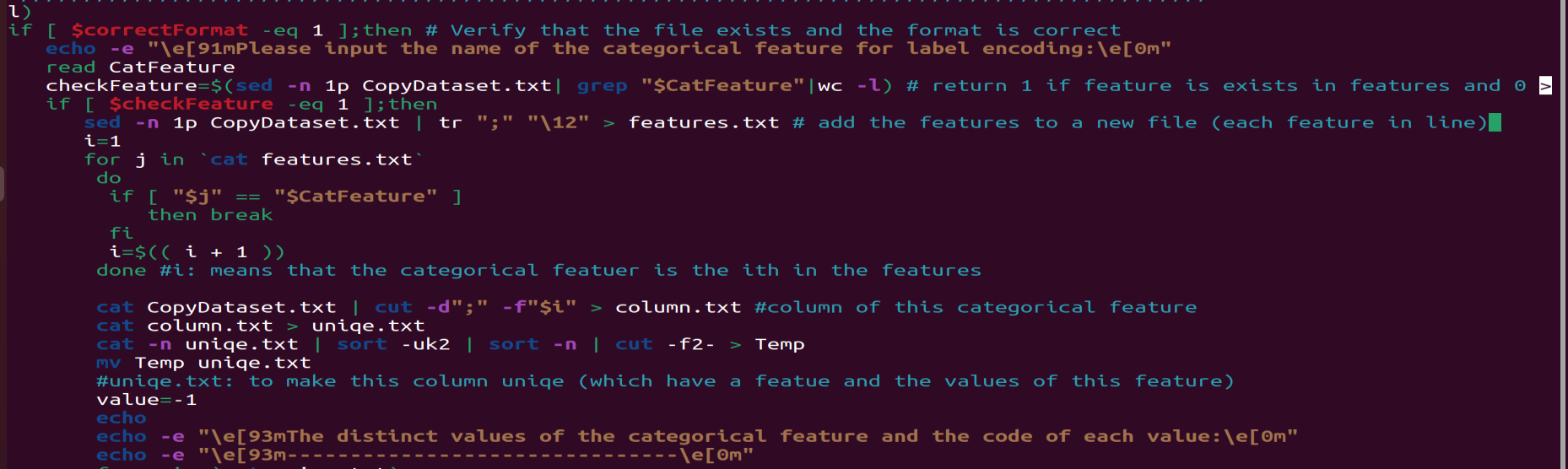


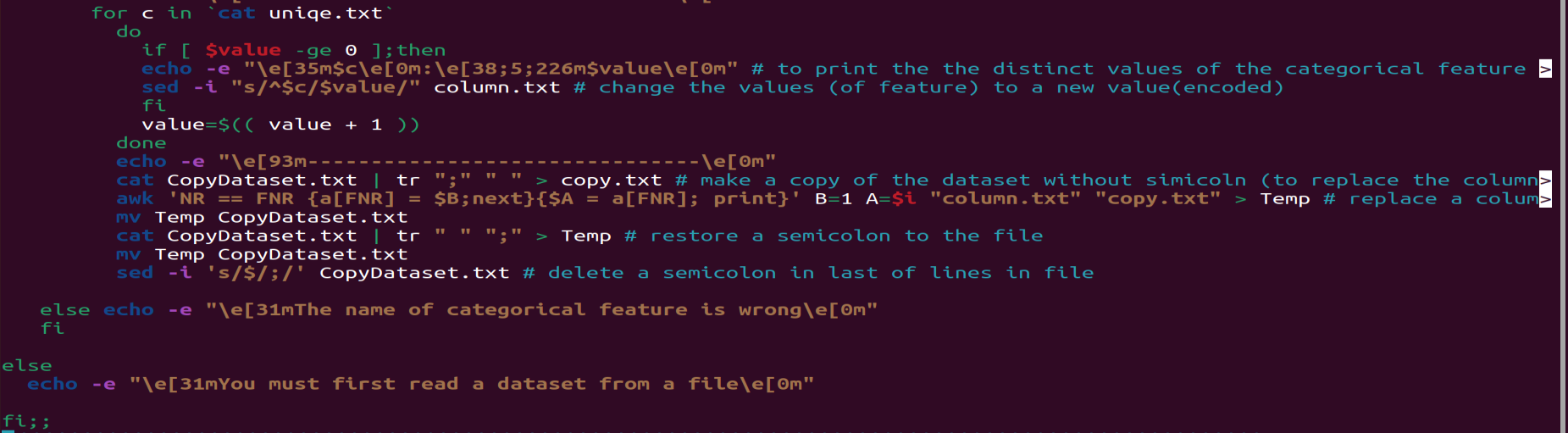
After one-hot encoding for the smoke feature, the names of all features of the dataset becomes :



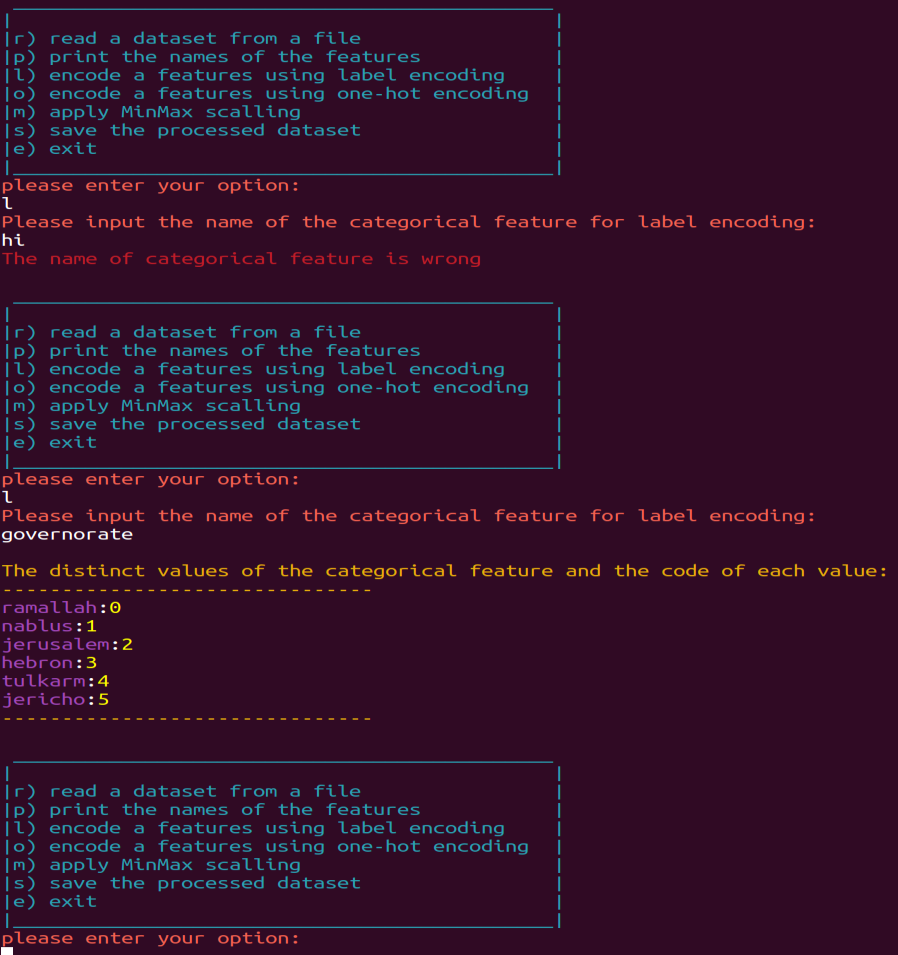
Procedure 4, when the user enters option “l”:

Code :

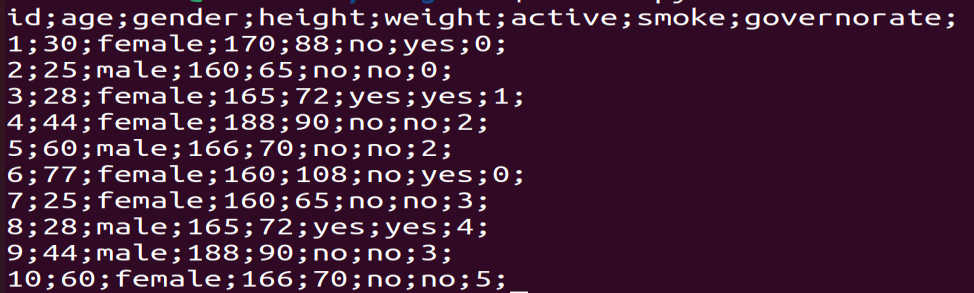




Description: The program must check if the categorical feature exists in dataset, and while there, It will prints on the screen the distinct values of the categorical feature and the code of each value. Then the categorical feature in the dataset will be encoded using label encoding . Otherwise it will prints on the screen “The name of categorical feature is wrong” .

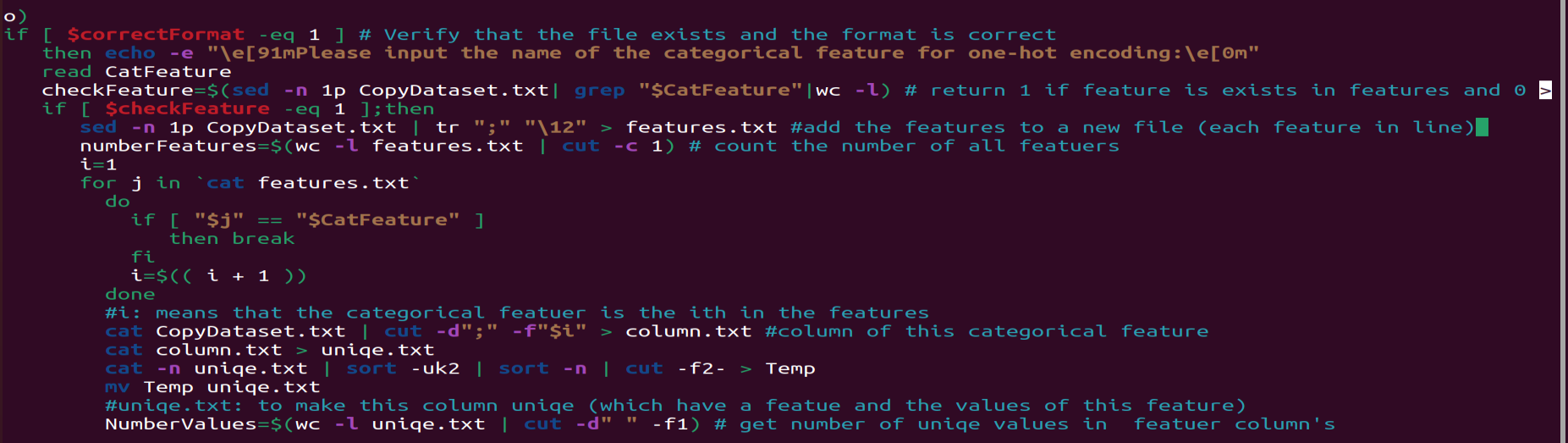
Output:

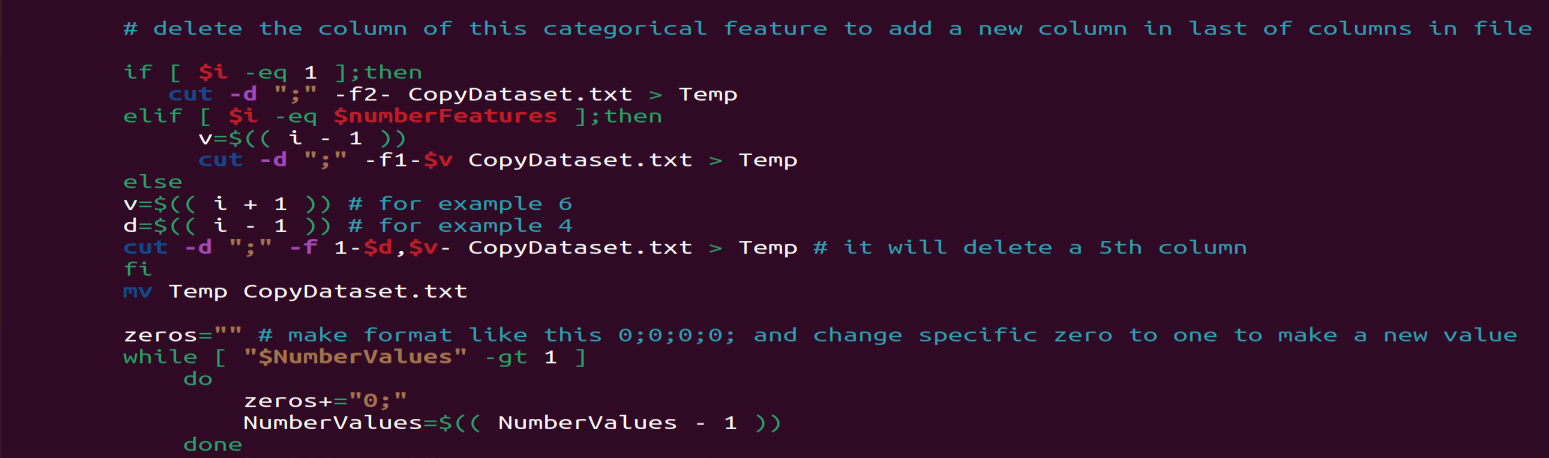
The file after doing the label encoding for governorate feature:

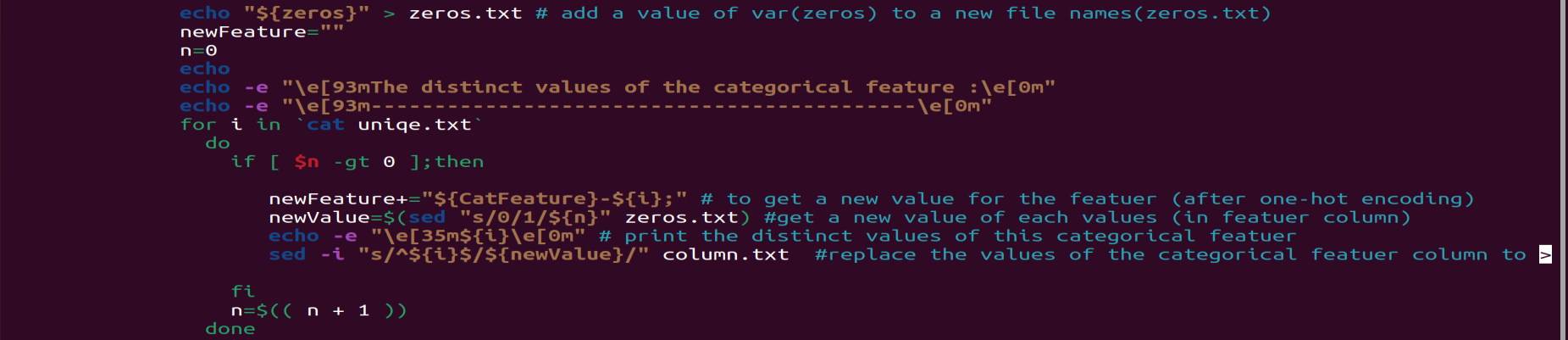


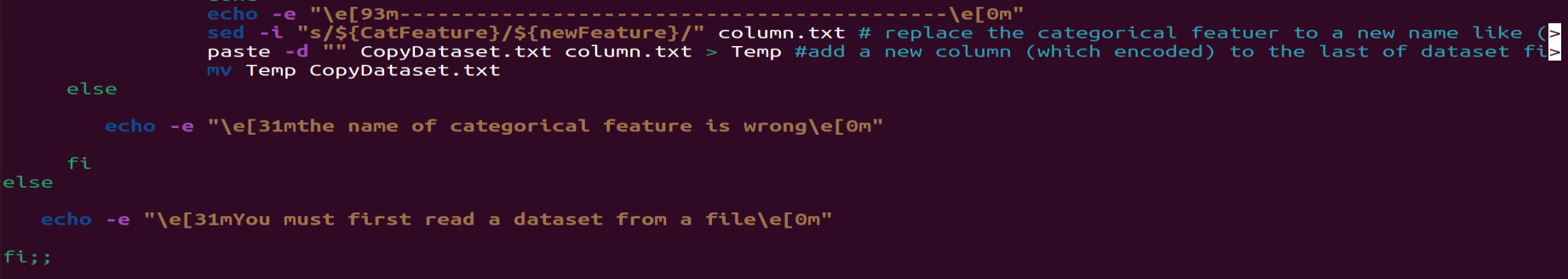
Procedure 5, when the user enters option “o”:

Code:





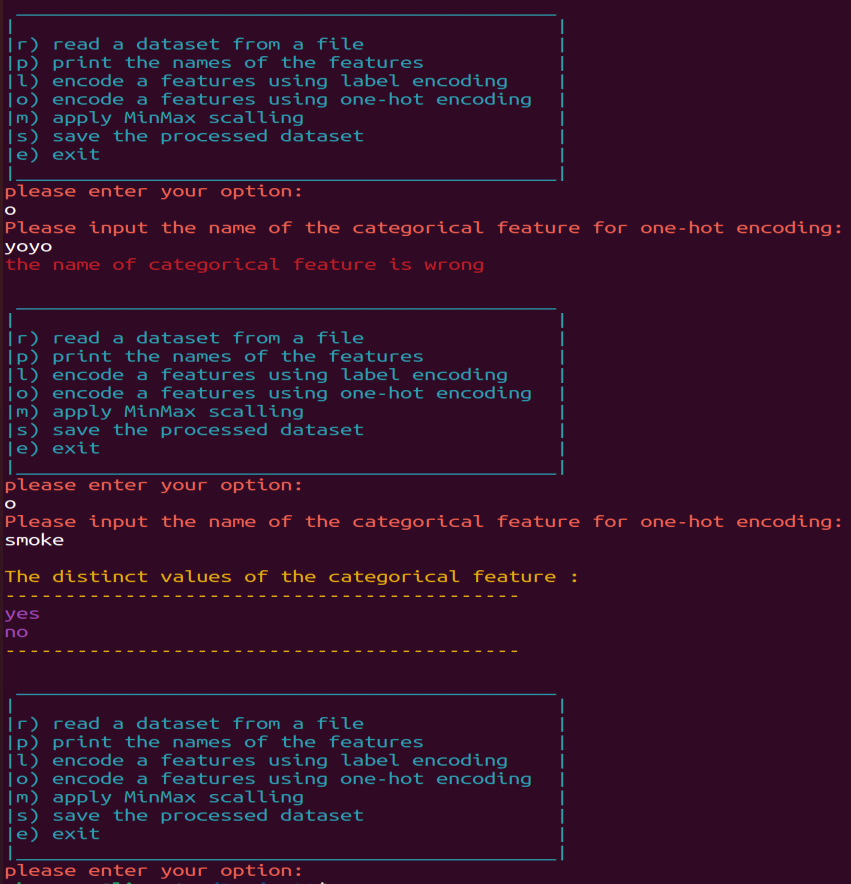




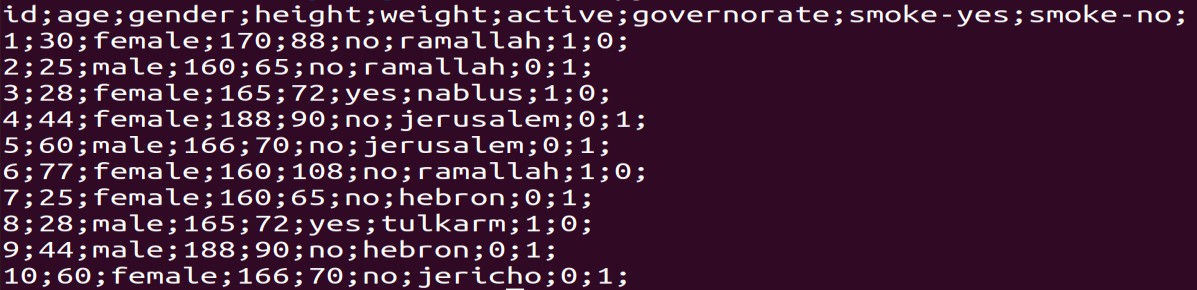
Description:

The program must check if the categorical feature exists in dataset, and while there, It will prints on the screen the distinct values of the categorical feature. Then the categorical feature in the dataset will be encoded using one-hot encoding. Otherwise it will prints on the screen “The name of categorical feature is wrong” .

Output:

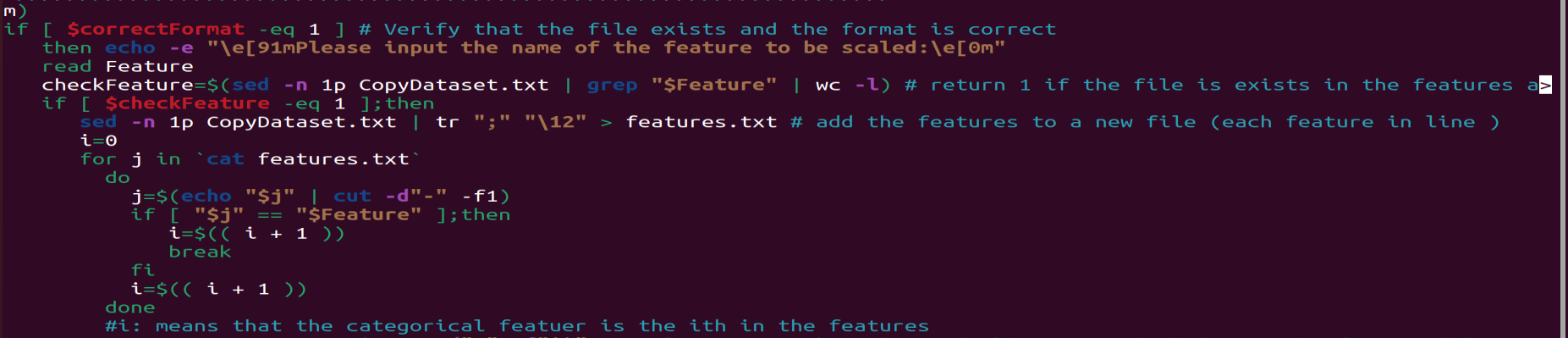


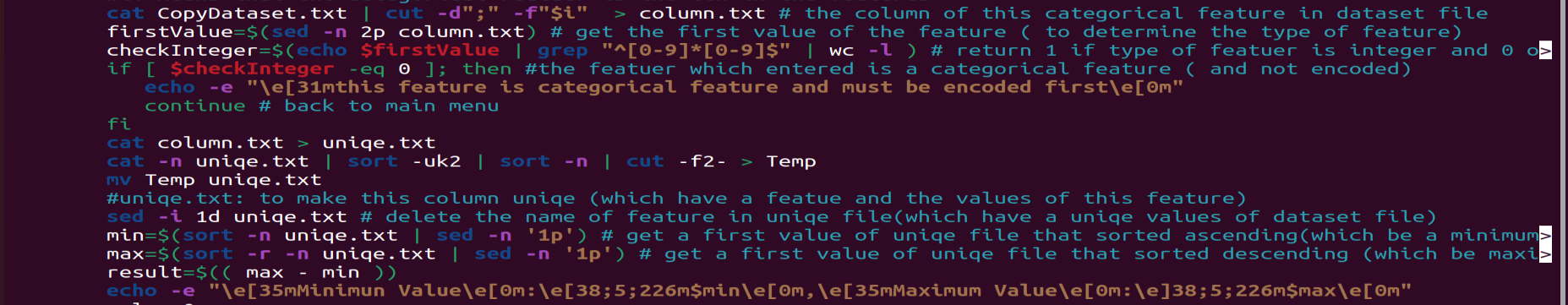
The file after doing the one-hot encoding for somke feature:

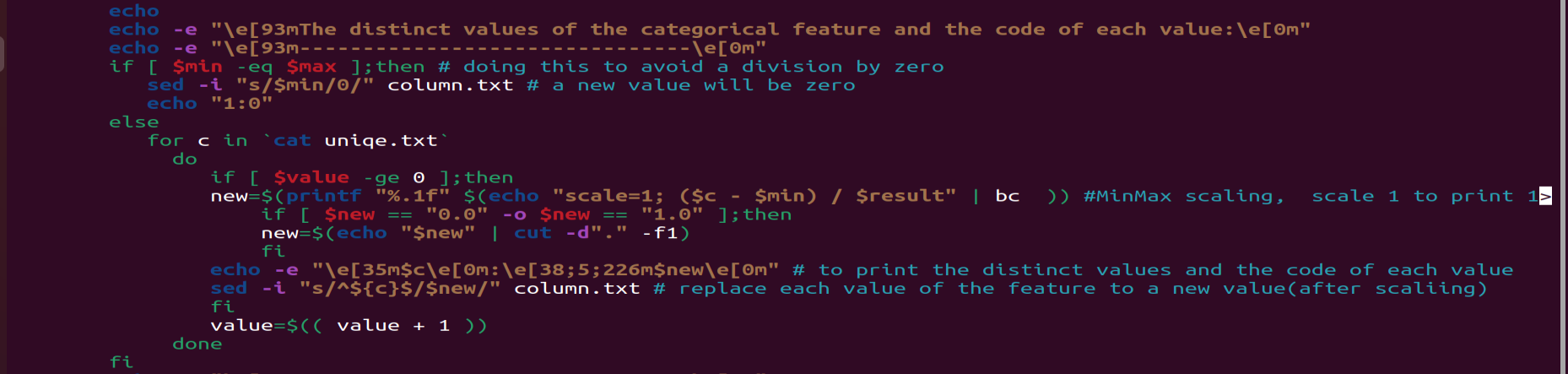


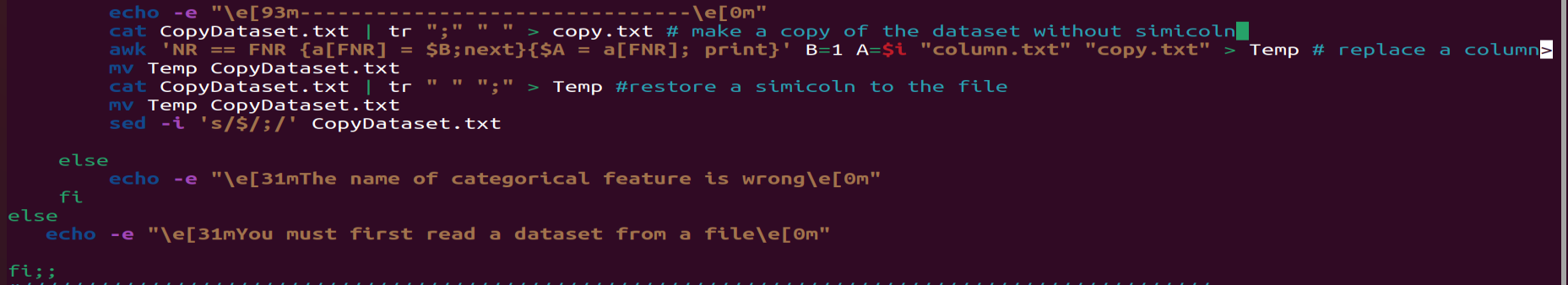
Procedure 6, when the user enters option “m”:

Code:





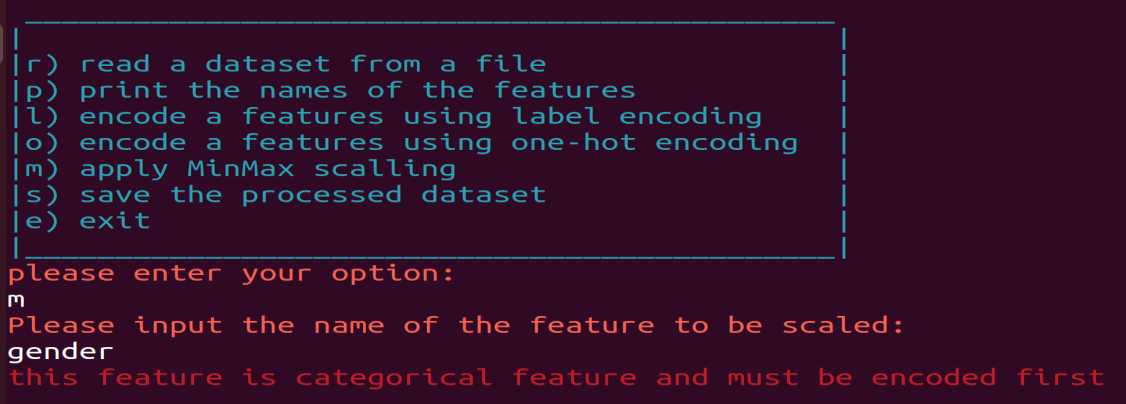


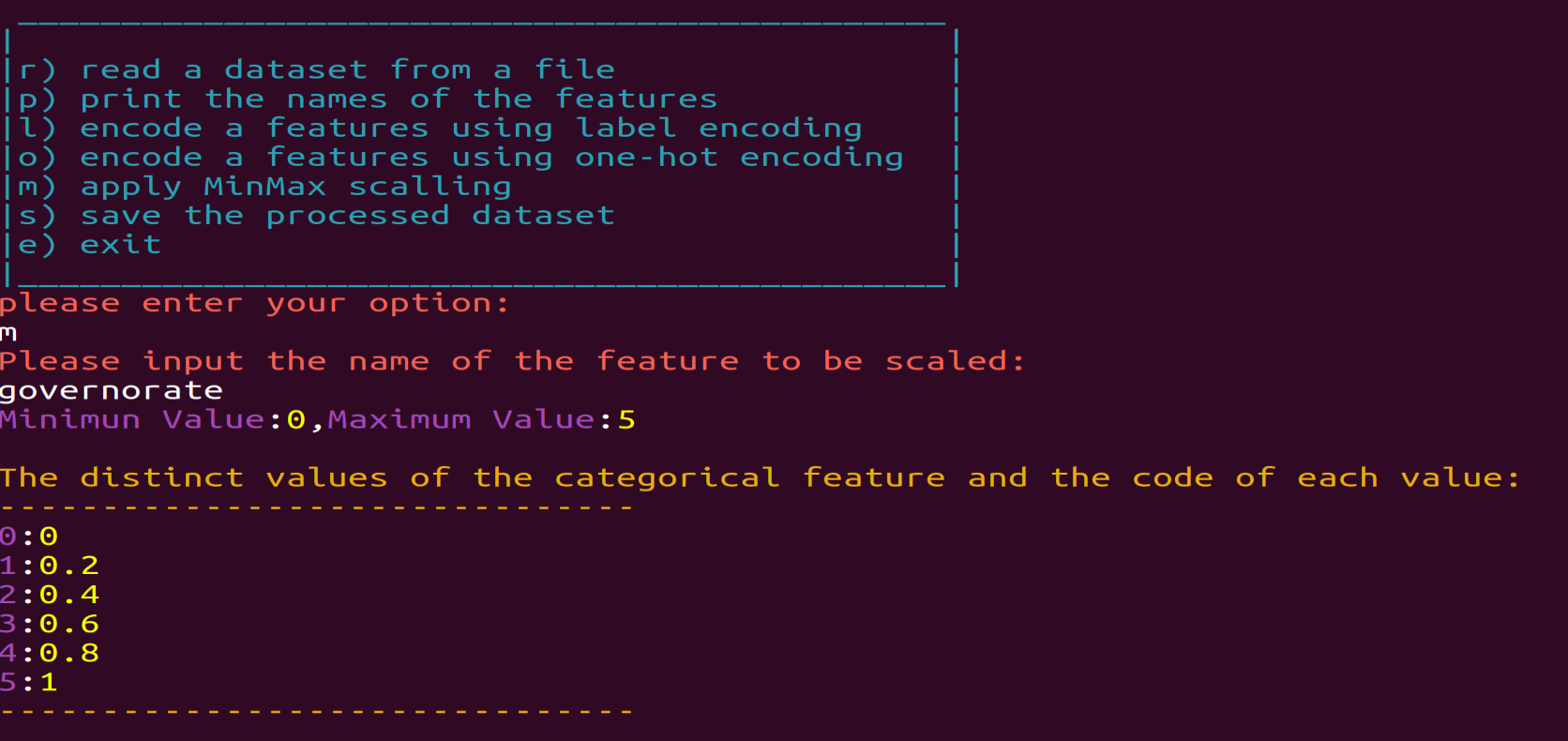


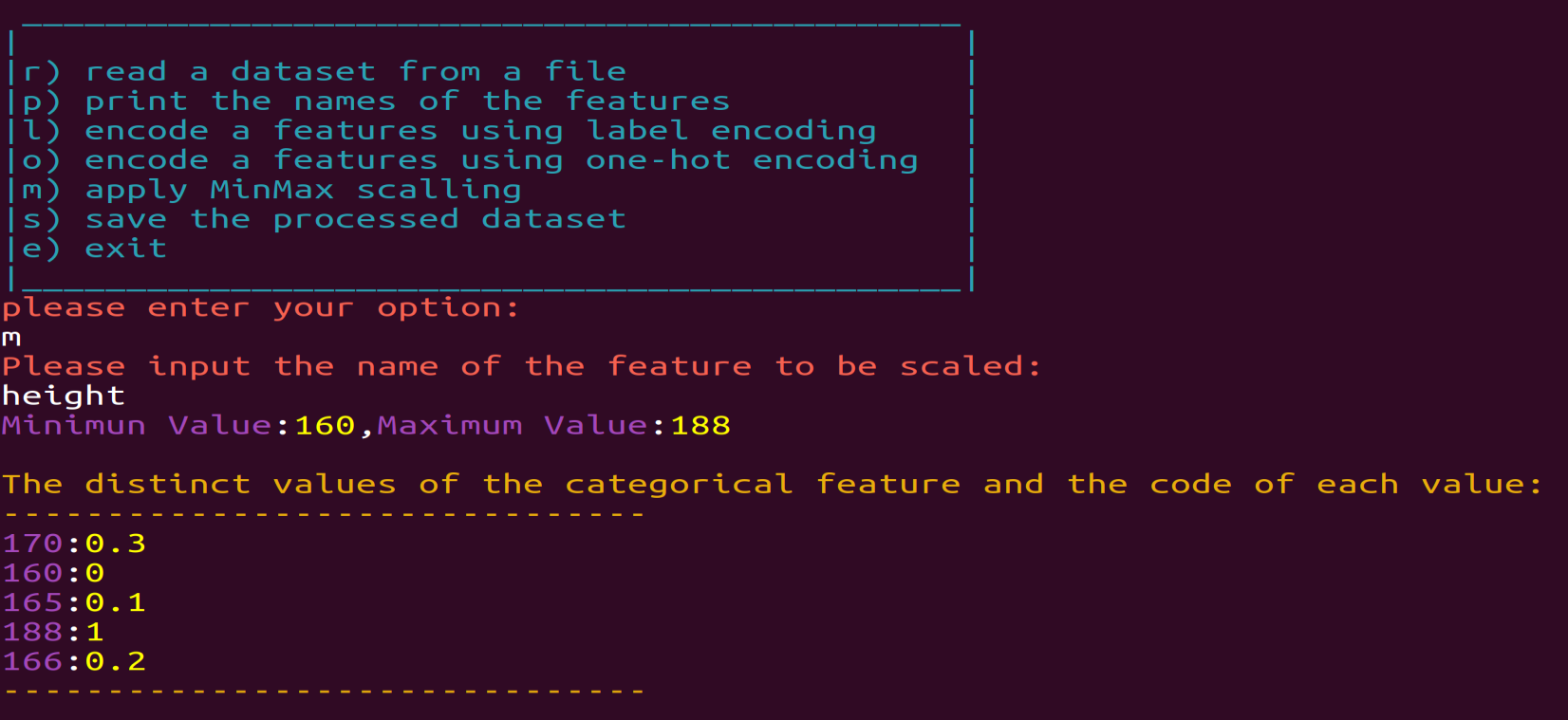
Description:

The program must check if the entered feature exists in the dataset, and while there, Then if the user entered a categorical feature, it must check if this feature is encoded, if the categorical feature didn’t encodes it will prints on the screen “this feature is categorical feature and must be encoded first”. And while this is encoded it will prints on the screen the minimum, maximum values of the feature , the distinct values of the feature and the code of each value . Then the feature in the dataset will be scaled using MinMax scaling. Otherwise it will prints on the screen “The name of categorical feature is wrong”.

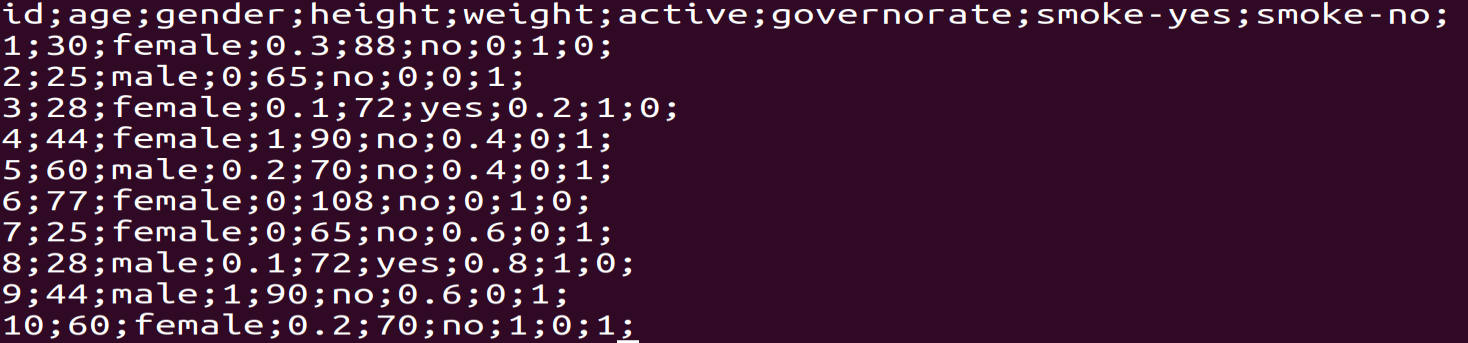
Output:





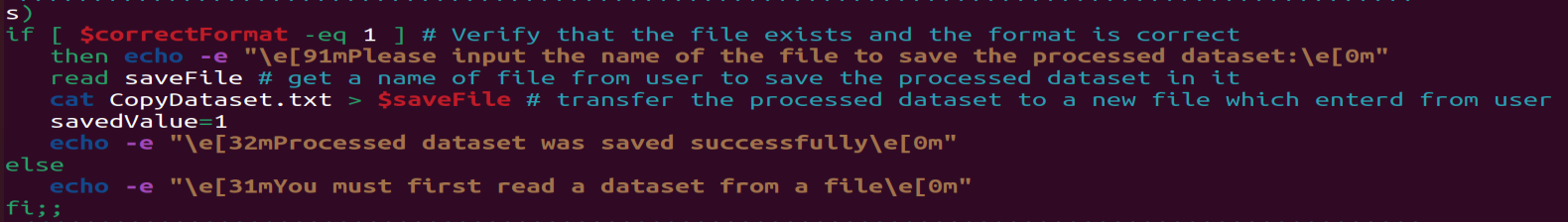


The file after doing the MinMax scaling for the governorate feature and the height feature:



Procedure 7, when the user enters option “s”:

Code:

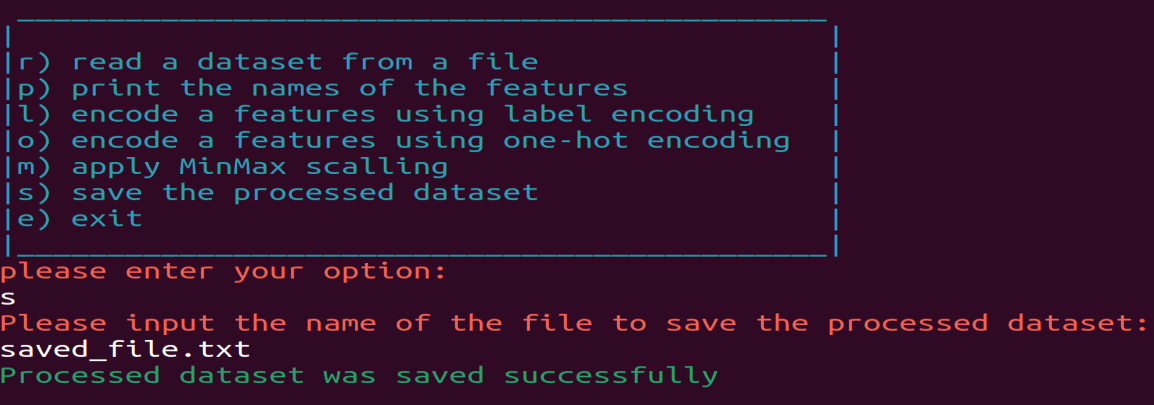


Description:

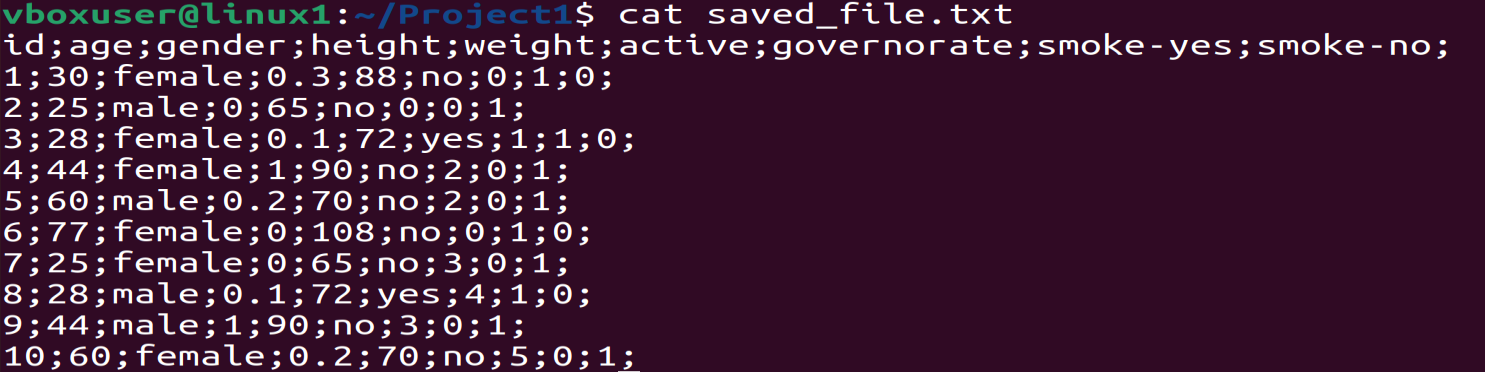
The program asks the user to enter the name of the file in which he wants to save the processed data set. Then it transfers the processed data set to the entered file name.

Changing the value of the "saveValue" variable to 1 means that the processed data has been saved.

Output:

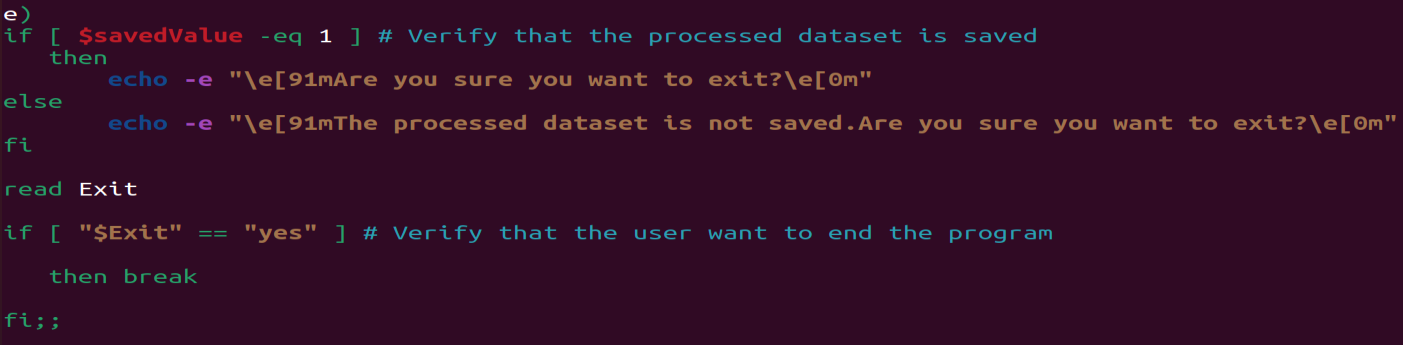


After saving the processed data set in saved\_file.txt :



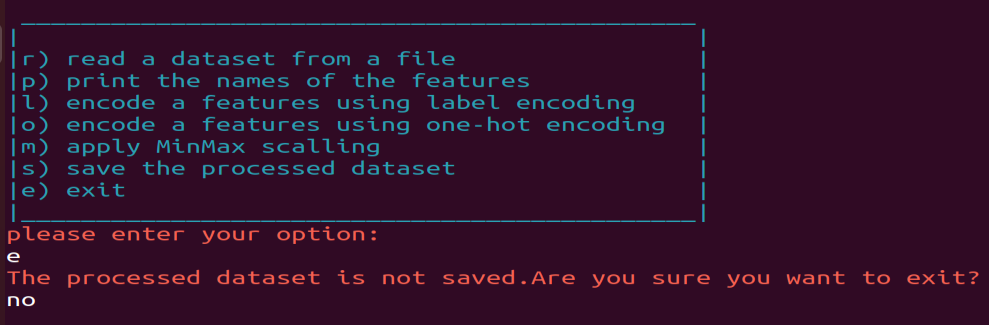
Procedure 8, when the user enters option “e”:

Code:

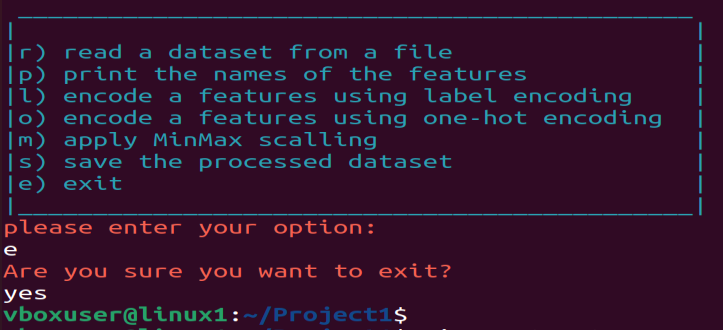


Output:

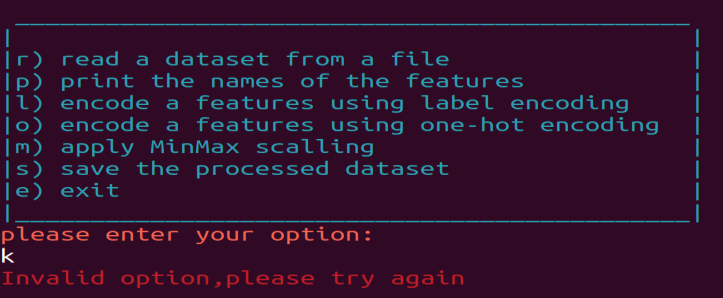
-When the processed data set is not saved in any file:



-When the processed data set is saved and the user wants to exit:

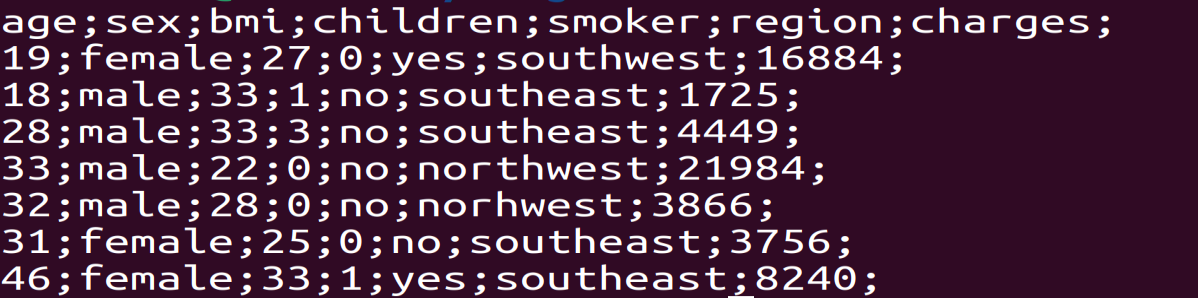


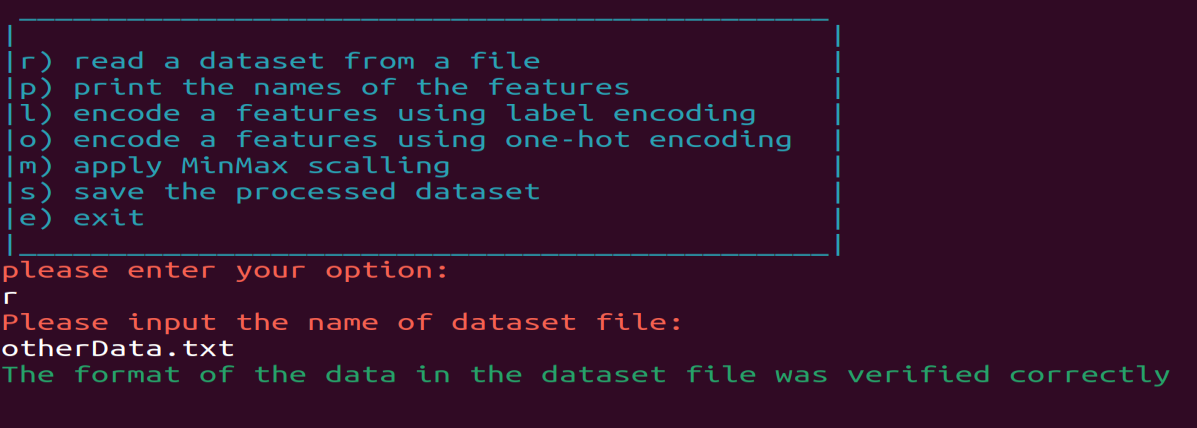
When the user entered an option that was not in the menu:

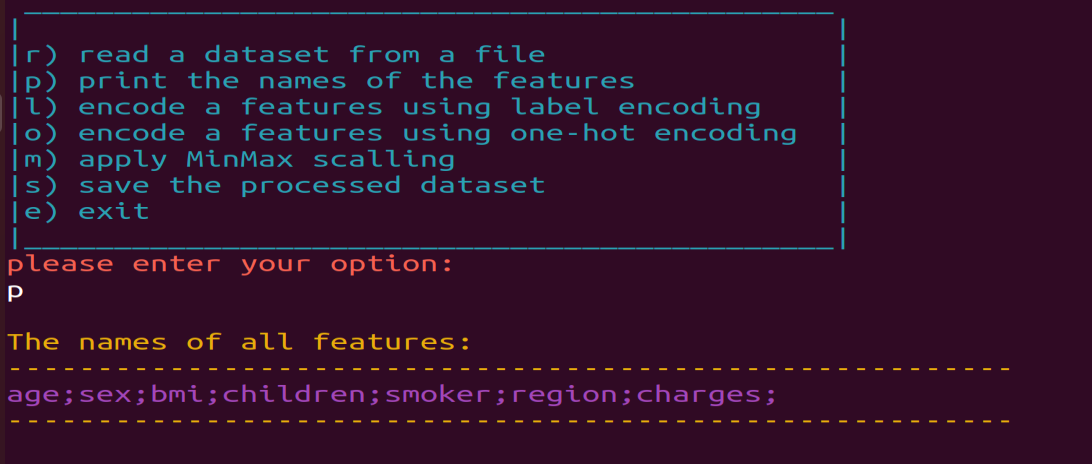


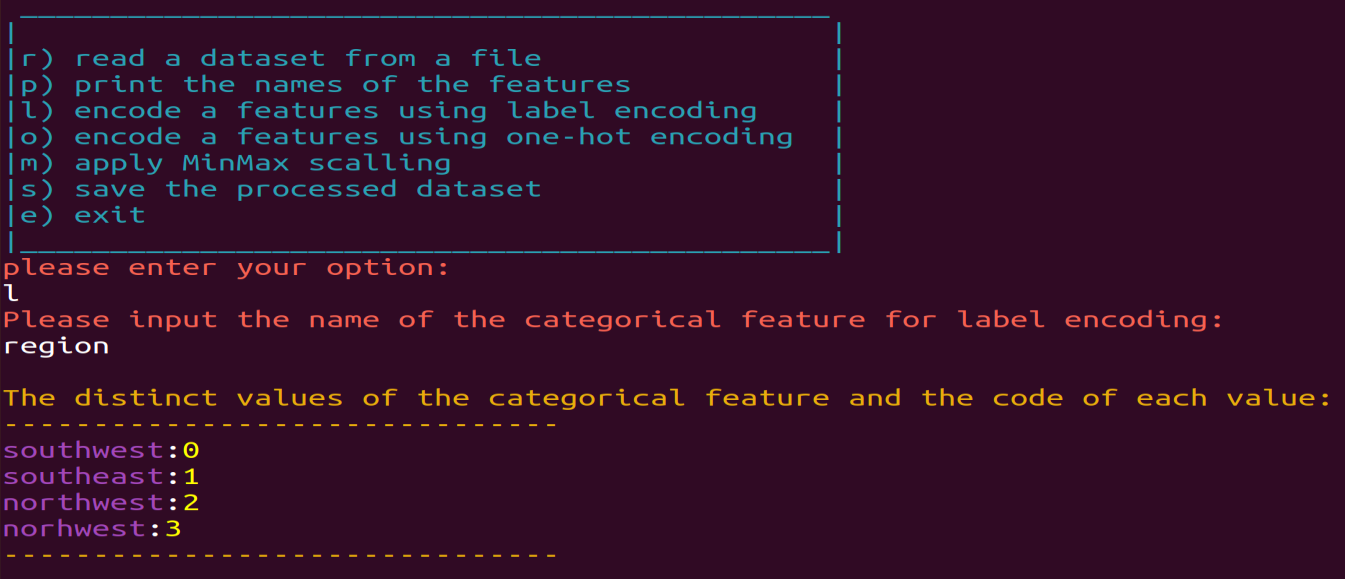
Another example of the dataset :

The File:

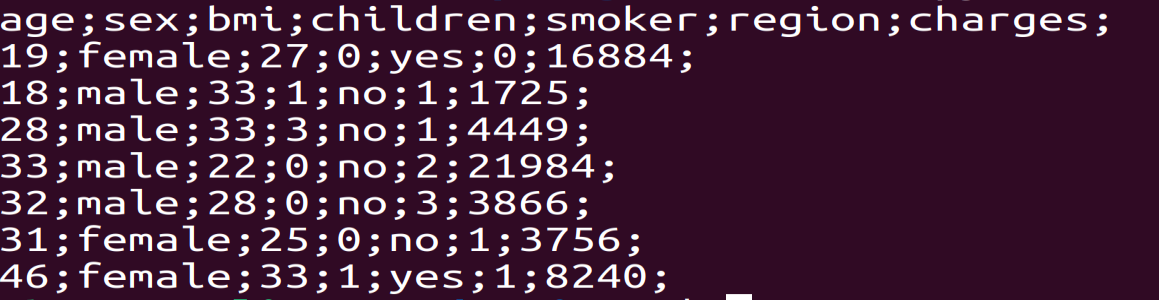


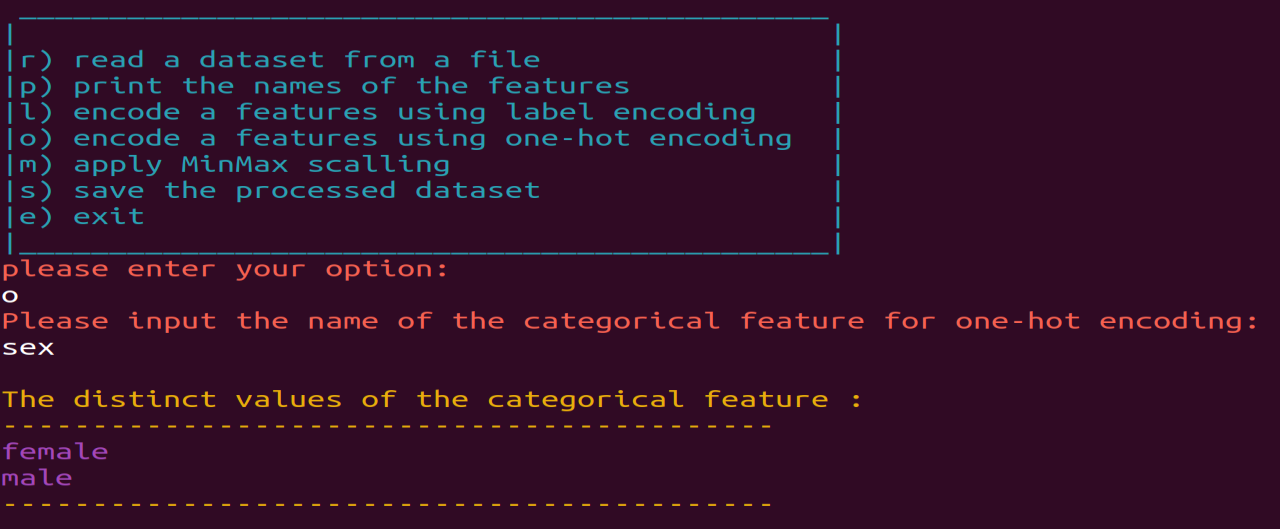




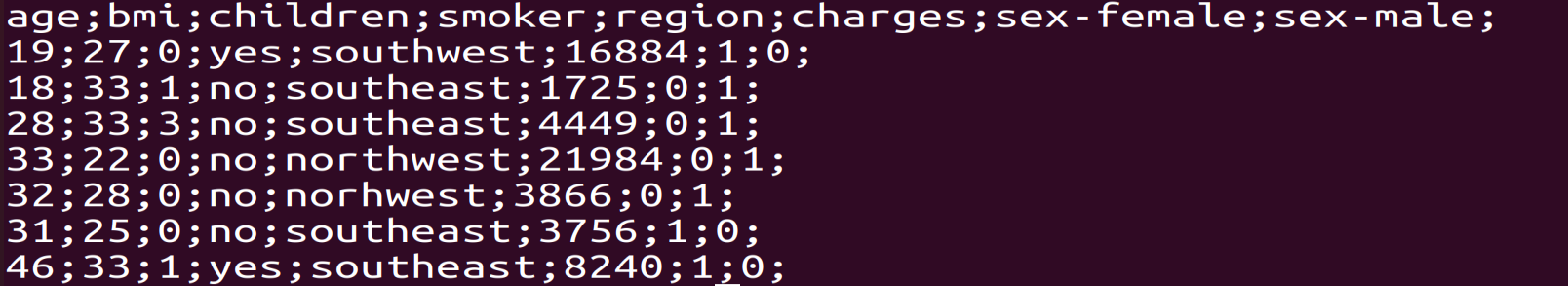


The file after doing the label encoding for region feature:

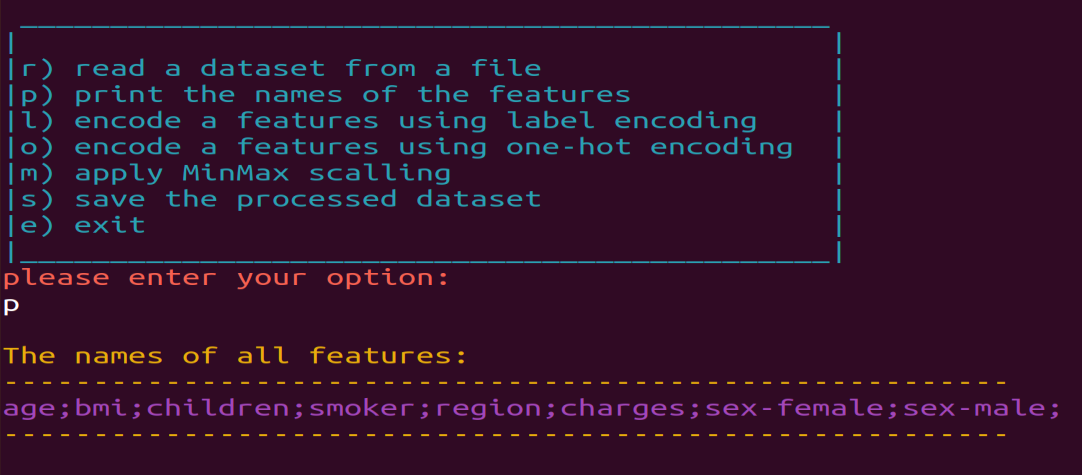


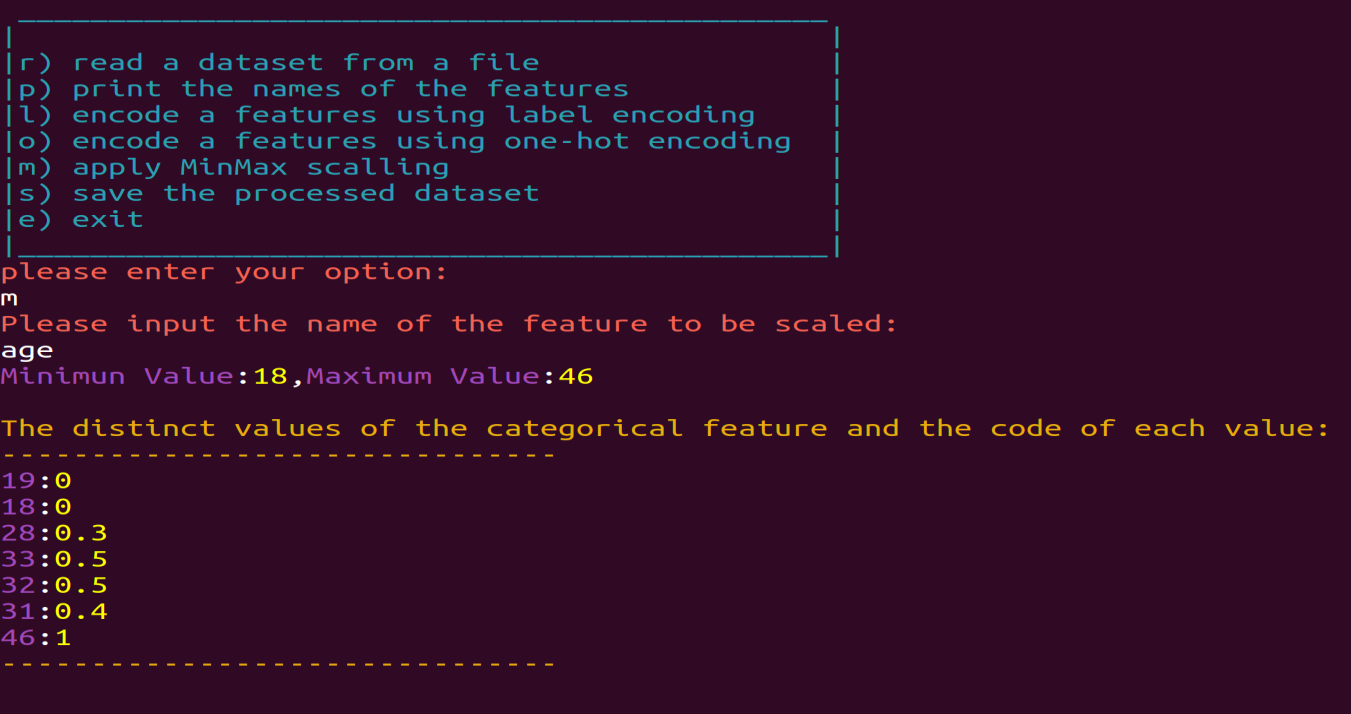


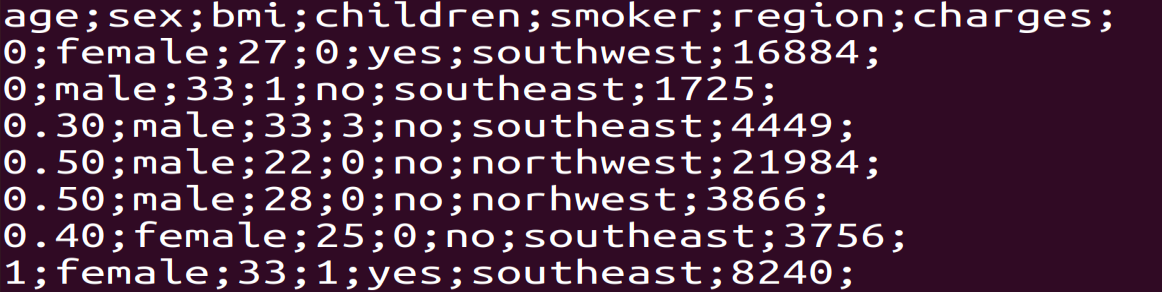
The file after doing the one-hot encoding for sex feature:

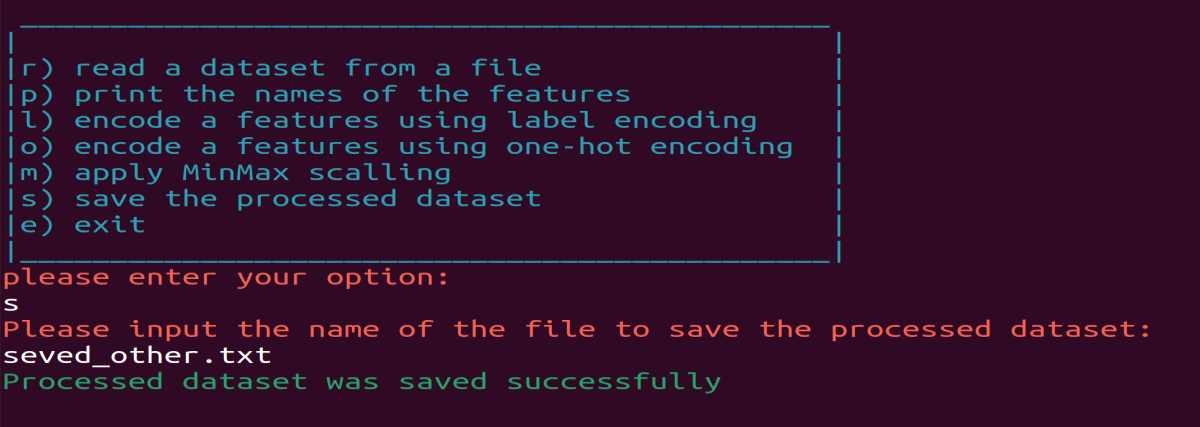


The names of features in the file after doing the one-hot encoding for sex feature:

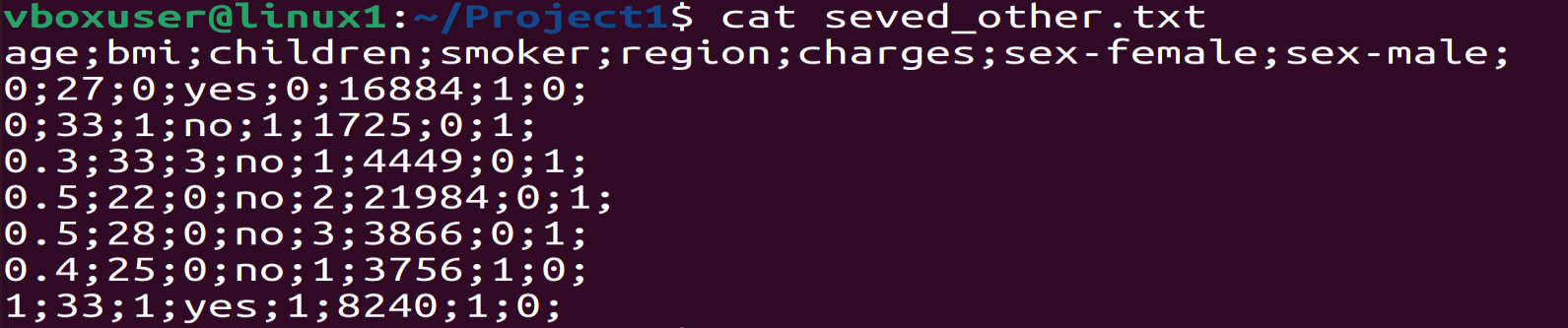




The file after doing the MinMax scaling for age feature:



After saving the processed data set in saved\_other.txt :



-When the processed data set is saved and the user wants to exit:



Full program code:

#Welcome to my project

# Osaid Hamza - 1200875

# Shell Project

savedValue=0

correctFormat=0

while true

do

echo

echo -e "\e[36m \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\e[0m"

echo -e "\e[36m| |\e[0m"

echo -e "\e[36m|r) read a dataset from a file |\e[0m"

echo -e "\e[36m|p) print the names of the features |\e[0m"

echo -e "\e[36m|l) encode a features using label encoding |\e[0m"

echo -e "\e[36m|o) encode a features using one-hot encoding |\e[0m"

echo -e "\e[36m|m) apply MinMax scalling |\e[0m"

echo -e "\e[36m|s) save the processed dataset |\e[0m"

echo -e "\e[36m|e) exit |\e[0m"

echo -e "\e[36m|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\e[0m"

echo -e "\e[91mplease enter your option:\e[0m"

read option

case $option in

#//////////////////////////////////////////////////////////////////////////////////////////////

r) echo -e "\e[91mPlease input the name of dataset file:\e[0m"

read FileName

if [ -e $FileName ];then # Verify that the file exists

firstLine=$(sed -n 1p dataset.txt) # get all features (in first line)

secondLine=$(sed -n 2p dataset.txt) # get all values of features

a=$(echo $firstLine | tr ";" "\n" | wc -l) # count number of the features

b=$(echo $secondLine | tr ";" "\n" | wc -l) # count number of the values

if [ $a -eq $b ];then # Verify that the number of features equals the number of values (that mean each feature taked a value,then the format is correct )

echo -e "\e[32mThe format of the data in the dataset file was verified correctly\e[0m"

correctFormat=1

savedValue=0

cat "$FileName" > CopyDataset.txt

else

echo -e "\e[31mThe format of the data in the dataset file is wrong\e[0m"

correctFormat=0

fi

else

echo -e "\e[31mfile does not exist\e[0m"

correctFormat=0

fi;;

#//////////////////////////////////////////////////////////////////////////////////////////////

p) if [ $correctFormat -eq 1 ];then # Verify that the file exists and the format is correct

echo

echo -e "\e[93mThe names of all features:\e[0m"

echo -e "\e[93m--------------------------------------------------------\e[0m"

echo -e "\e[35m`sed -n '1p' CopyDataset.txt` \e[0m" # print the first line in the file (The Features)

echo -e "\e[93m--------------------------------------------------------\e[0m"

else

echo -e "\e[31mYou must first read a dataset from a file\e[0m"

fi;;

#//////////////////////////////////////////////////////////////////////////////////////////////

l)

if [ $correctFormat -eq 1 ];then # Verify that the file exists and the format is correct

echo -e "\e[91mPlease input the name of the categorical feature for label encoding:\e[0m"

read CatFeature

checkFeature=$(sed -n 1p CopyDataset.txt| grep "$CatFeature"|wc -l) # return 1 if feature is exists in features and 0 otherwise

if [ $checkFeature -eq 1 ];then

sed -n 1p CopyDataset.txt | tr ";" "\12" > features.txt # add the features to a new file (each feature in line)

i=1

for j in `cat features.txt`

do

if [ "$j" == "$CatFeature" ]

then break

fi

i=$(( i + 1 ))

done #i: means that the categorical featuer is the ith in the features

cat CopyDataset.txt | cut -d";" -f"$i" > column.txt #column of this categorical feature

cat column.txt > uniqe.txt

cat -n uniqe.txt | sort -uk2 | sort -n | cut -f2- > Temp

mv Temp uniqe.txt

#uniqe.txt: to make this column uniqe (which have a featue and the values of this feature)

value=-1

echo

echo -e "\e[93mThe distinct values of the categorical feature and the code of each value:\e[0m"

echo -e "\e[93m-------------------------------\e[0m"

for c in `cat uniqe.txt`

do

if [ $value -ge 0 ];then

echo -e "\e[35m$c\e[0m:\e[38;5;226m$value\e[0m" # to print the the distinct values of the categorical feature and the code of each value

sed -i "s/^$c/$value/" column.txt # change the values (of feature) to a new value(encoded)

fi

value=$(( value + 1 ))

done

echo -e "\e[93m-------------------------------\e[0m"

cat CopyDataset.txt | tr ";" " " > copy.txt # make a copy of the dataset without simicoln (to replace the column of the categorical feature which entered to a new column (which have a new values (encoded) )

awk 'NR == FNR {a[FNR] = $B;next}{$A = a[FNR]; print}' B=1 A=$i "column.txt" "copy.txt" > Temp # replace a column of the feature (i) to a new column (B=1->first column in column.txt)

mv Temp CopyDataset.txt

cat CopyDataset.txt | tr " " ";" > Temp # restore a semicolon to the file

mv Temp CopyDataset.txt

sed -i 's/$/;/' CopyDataset.txt # delete a semicolon in last of lines in file

else echo -e "\e[31mThe name of categorical feature is wrong\e[0m"

fi

else

echo -e "\e[31mYou must first read a dataset from a file\e[0m"

fi;;

#//////////////////////////////////////////////////////////////////////////////////////////////////

o)

if [ $correctFormat -eq 1 ] # Verify that the file exists and the format is correct

then echo -e "\e[91mPlease input the name of the categorical feature for one-hot encoding:\e[0m"

read CatFeature

checkFeature=$(sed -n 1p CopyDataset.txt| grep "$CatFeature"|wc -l) # return 1 if feature is exists in features and 0 otherwise

if [ $checkFeature -eq 1 ];then

sed -n 1p CopyDataset.txt | tr ";" "\12" > features.txt #add the features to a new file (each feature in line)

numberFeatures=$(wc -l features.txt | cut -c 1) # count the number of all featuers

i=1

for j in `cat features.txt`

do

if [ "$j" == "$CatFeature" ]

then break

fi

i=$(( i + 1 ))

done

#i: means that the categorical featuer is the ith in the features

cat CopyDataset.txt | cut -d";" -f"$i" > column.txt #column of this categorical feature

cat column.txt > uniqe.txt

cat -n uniqe.txt | sort -uk2 | sort -n | cut -f2- > Temp

mv Temp uniqe.txt

#uniqe.txt: to make this column uniqe (which have a featue and the values of this feature)

NumberValues=$(wc -l uniqe.txt | cut -d" " -f1) # get number of uniqe values in featuer column's

# delete the column of this categorical feature to add a new column in last of columns in file

if [ $i -eq 1 ];then

cut -d ";" -f2- CopyDataset.txt > Temp

elif [ $i -eq $numberFeatures ];then

v=$(( i - 1 ))

cut -d ";" -f1-$v CopyDataset.txt > Temp

else

v=$(( i + 1 )) # for example 6

d=$(( i - 1 )) # for example 4

cut -d ";" -f 1-$d,$v- CopyDataset.txt > Temp # it will delete a 5th column

fi

mv Temp CopyDataset.txt

zeros="" # make format like this 0;0;0;0; and change specific zero to one to make a new value

while [ "$NumberValues" -gt 1 ]

do

zeros+="0;"

NumberValues=$(( NumberValues - 1 ))

done

echo "${zeros}" > zeros.txt # add a value of var(zeros) to a new file names(zeros.txt)

newFeature=""

n=0

echo

echo -e "\e[93mThe distinct values of the categorical feature :\e[0m"

echo -e "\e[93m-------------------------------------------\e[0m"

for i in `cat uniqe.txt`

do

if [ $n -gt 0 ];then

newFeature+="${CatFeature}-${i};" # to get a new value for the featuer (after one-hot encoding)

newValue=$(sed "s/0/1/${n}" zeros.txt) #get a new value of each values (in featuer column)

echo -e "\e[35m${i}\e[0m" # print the distinct values of this categorical featuer

sed -i "s/^${i}$/${newValue}/" column.txt #replace the values of the categorical featuer column to a new value

fi

n=$(( n + 1 ))

done

echo -e "\e[93m-------------------------------------------\e[0m"

sed -i "s/${CatFeature}/${newFeature}/" column.txt # replace the categorical featuer to a new name like (featuer-firstValue;featuer-secondValue;...)

paste -d "" CopyDataset.txt column.txt > Temp #add a new column (which encoded) to the last of dataset file

mv Temp CopyDataset.txt

else

echo -e "\e[31mthe name of categorical feature is wrong\e[0m"

fi

else

echo -e "\e[31mYou must first read a dataset from a file\e[0m"

fi;;

#/////////////////////////////////////////////////////////////////////

m)

if [ $correctFormat -eq 1 ] # Verify that the file exists and the format is correct

then echo -e "\e[91mPlease input the name of the feature to be scaled:\e[0m"

read Feature

checkFeature=$(sed -n 1p CopyDataset.txt | grep "$Feature" | wc -l) # return 1 if the file is exists in the features and 0 otherwise

if [ $checkFeature -eq 1 ];then

sed -n 1p CopyDataset.txt | tr ";" "\12" > features.txt # add the features to a new file (each feature in line )

i=0

for j in `cat features.txt`

do

j=$(echo "$j" | cut -d"-" -f1)

if [ "$j" == "$Feature" ];then

i=$(( i + 1 ))

break

fi

i=$(( i + 1 ))

done

#i: means that the categorical featuer is the ith in the features

cat CopyDataset.txt | cut -d";" -f"$i" > column.txt # the column of this categorical feature in dataset file

firstValue=$(sed -n 2p column.txt) # get the first value of the feature ( to determine the type of feature)

checkInteger=$(echo $firstValue | grep "^[0-9]\*[0-9]$" | wc -l ) # return 1 if type of featuer is integer and 0 otherwise

if [ $checkInteger -eq 0 ]; then #the featuer which entered is a categorical feature ( and not encoded)

echo -e "\e[31mthis feature is categorical feature and must be encoded first\e[0m"

continue # back to main menu

fi

cat column.txt > uniqe.txt

cat -n uniqe.txt | sort -uk2 | sort -n | cut -f2- > Temp

mv Temp uniqe.txt

#uniqe.txt: to make this column uniqe (which have a featue and the values of this feature)

sed -i 1d uniqe.txt # delete the name of feature in uniqe file(which have a uniqe values of dataset file)

min=$(sort -n uniqe.txt | sed -n '1p') # get a first value of uniqe file that sorted ascending(which be a minimum value)

max=$(sort -r -n uniqe.txt | sed -n '1p') # get a first value of uniqe file that sorted descending (which be maximum value)

result=$(( max - min ))

echo -e "\e[35mMinimun Value\e[0m:\e[38;5;226m$min\e[0m,\e[35mMaximum Value\e[0m:\e[38;5;226m$max\e[0m"

value=0

echo

echo -e "\e[93mThe distinct values of the categorical feature and the code of each value:\e[0m"

echo -e "\e[93m-------------------------------\e[0m"

if [ $min -eq $max ];then # doing this to avoid a division by zero

sed -i "s/$min/0/" column.txt # a new value will be zero

echo "1:0"

else

for c in `cat uniqe.txt`

do

if [ $value -ge 0 ];then

new=$(printf "%.2f" $(echo "scale=1; ($c - $min) / $result" | bc )) #MinMax scaling, scale 1 to print 1 digit after dot

if [ $new == "0.00" -o $new == "1.00" ];then

new=$(echo "$new" | cut -d"." -f1)

fi

echo -e "\e[35m$c\e[0m:\e[38;5;226m$new\e[0m" # to print the distinct values and the code of each value

sed -i "s/^${c}$/$new/" column.txt # replace each value of the feature to a new value(after scaliing)

fi

value=$(( value + 1 ))

done

fi

echo -e "\e[93m-------------------------------\e[0m"

cat CopyDataset.txt | tr ";" " " > copy.txt # make a copy of the dataset without simicoln

awk 'NR == FNR {a[FNR] = $B;next}{$A = a[FNR]; print}' B=1 A=$i "column.txt" "copy.txt" > Temp # replace a column of the feature (i) to a new column (B=1->first column in column.txt)

mv Temp CopyDataset.txt

cat CopyDataset.txt | tr " " ";" > Temp #restore a simicoln to the file

mv Temp CopyDataset.txt

sed -i 's/$/;/' CopyDataset.txt

else

echo -e "\e[31mThe name of categorical feature is wrong\e[0m"

fi

else

echo -e "\e[31mYou must first read a dataset from a file\e[0m"

fi;;

#//////////////////////////////////////////////////////////////////////////////////////////////

s)

if [ $correctFormat -eq 1 ] # Verify that the file exists and the format is correct

then echo -e "\e[91mPlease input the name of the file to save the processed dataset:\e[0m"

read saveFile # get a name of file from user to save the processed dataset in it

cat CopyDataset.txt > $saveFile # transfer the processed dataset to a new file which enterd from user

savedValue=1

echo -e "\e[32mProcessed dataset was saved successfully\e[0m"

else

echo -e "\e[31mYou must first read a dataset from a file\e[0m"

fi;;

#//////////////////////////////////////////////////////////////////////////////////////////////

e)

if [ $savedValue -eq 1 ] # Verify that the processed dataset is saved

then

echo -e "\e[91mAre you sure you want to exit?\e[0m"

else

echo -e "\e[91mThe processed dataset is not saved.Are you sure you want to exit?\e[0m"

fi

read Exit

if [ "$Exit" == "yes" ] # Verify that the user want to end the program

then break

fi;;

#//////////////////////////////////////////////////////////////////////////////////////////////

\*) echo -e "\e[31mInvalid option,please try again\e[0m"

esac

done