

**Faculty of Engineering & Technology**

**Electrical & Computer Engineering Department**

**ENCS3130**

**Report # 1**

**SHELL PROJECT**

**Prepared by :**

Ahmad Ismail 1202450

**Instructor :** Dr. Aziz Qaroush

**Section :** 4

**Date:**1st-Jan-2023

Code :

status=0

dataSaved=0

tempFile=datasetTEMP.txt

while true

do

echo "select one from the list"

echo "r) Read a dataset from a file "

echo "p) print the names of the features "

echo "l) encode feature using label encoding "

echo "o) encode feature using one-hot encoding "

echo "m) apply MinMax scalling"

echo "s) save the processed dataset "

echo "e) Exit "

read -p " --------------------> : " choice

case "$choice" in

r)

read -p "Please input the name of the dataset file (do not include .txt) : " fileName # read the file name from user

fileName=$fileName.txt

if [ -e $fileName ] # checks if file exists

then

fs1=$( awk -F ";" 'NR ==1 {print NF }' $fileName ) # get num of data fields

fs2=$( awk -F ";" 'NR ==2 {print NF }' $fileName ) # get num of data fields

if [ $fs1 -eq $fs2 ] # to check if the file is clean or not

then

echo

clear

echo "file is read successfully :)"

echo

echo "------------------------------ FILE Features ------------------------------"

fs=$(awk -F ";" 'NR==1{print $0}' $fileName | sed 's/;/ /g')

echo $fs

echo "-------------------------------- DONE ---------------------------------"

cp $fileName $tempFile # making temporary dataset file to work on

status=1

else

echo "file is not clean"

fi

else

clear

echo "file with name ($fileName) does not exist :( "

echo

echo

fi

echo Back to main menu ...

echo

echo

sleep 2

;;

p)

if [ $status -ne 0 ] # to check if the file readed

then

clear

echo

echo "------------------------------ FILE DATA ------------------------------"

echo

awk -F ";" '{$1=""; print $0}' $tempFile # print the data in the file

echo

echo "-------------------------------- DONE ---------------------------------"

echo

sleep 2

else

echo "Please read the file first"

sleep 1

echo Back to main menu ...

sleep 1

clear

fi

;;

l)

if [ $status -ne 0 ] # to check if the file readed

then

read -p " Please input the name of the categorical feature for label encoding : " feature

# read the feature name from user - p to read from same line

fieldNum=$( awk -v b="$feature" -F ";" 'NR==1 {for(i=1;i<=NF;i++){if($i ~ b ){print i}}}' $tempFile) #saves field num

# in awk -v --> letting me to use variables in awk cuz of '$var'

if [ -z "$fieldNum" ] # to check if we found the feature or not

then # the feature doesn't exists

clear

echo

echo "The name of categorical feature ($feature) is wrong :("

echo

echo Back to main menu ...

sleep 2

else # found it maybe

featureName=$(awk -v f=$fieldNum -F ";" ' NR==1 {print $f}' $tempFile ) # saves feature name from dataset

if [ "$feature" == "$featureName" ] # to check if the field have same feature name

then # its the same

(awk -v f=$fieldNum -F ";" ' NR>1 {print $f}' $tempFile > featureValuesTemp.txt )

#NR>1 --> to skip the first line cuz it contains the feature name

# getting all feature inputs and add them to temp file to work on with

numOfEncodedStrings=$(wc -l featureValuesTemp.txt | awk -F " " '{ print $1 }')

# getting num of lines from featureValuesTemp.txt file ^^^

for ((i=0; i <= $numOfEncodedStrings ; i++))

# loop in temp file to check every feature input and add it to unique file if not added yet

do

featureTemp=$(awk -v I=$i -F " " 'NR == I { print $1 }' featureValuesTemp.txt) # first column data saves

# -F is unneciary cuz we have only one column in the file

touch featureValues.txt # to create the unique file of feature inputs

lineNum=$( grep -n -w "$featureTemp" featureValues.txt | awk -F ":" '{print $1}') #saves line num

# -n to print line number

if [ -z "$lineNum" ] # to check if we found the feature Value or not from unique file

then # when we cannot find the line num of it then the data is not added !!

echo $featureTemp >> featureValues.txt # then we will add it

lineNum=$( grep -n -w "$featureTemp" featureValues.txt | awk -F ":" '{print $1}') #saves field num

else

featureName=$(awk -v f=$lineNum -F " " ' NR==f {print $1}' featureValues.txt)

# gets data from featureValues.txt (that conatins feature inputs)

fi

done

echo

echo --------------------------------- LABEL ENCODING ---------------------------------

for ((i=2; i <= $numOfEncodedStrings ; i++)) # loop to print the feature inputs and their values

do

tmp=$(awk -v I=$i ' NR==I {print $1 }' featureValues.txt )

if [ -z "$tmp" ] # to check if we found the feature Value or not from unique file

then # when we cannot find the line num of it then the data is not added !!

break

else

echo "feature $tmp Value is : $i"

fi

done

echo "--------------------------------- feature Inputs ^^^ ---------------------------------"

numOfEncodedStrings=$(wc -l $tempFile | awk -F " " '{print $1}')

numOfEncodedStrings+=1 # to replace the last line

touch temp.txt

awk -F " " ' NR == 1 {print $1 }' $tempFile > temp.txt # to add the header to the temp file

for ((i=2; i <= $numOfEncodedStrings ; i++))

do

tmp=$(awk -v I=$i -v field=$fieldNum -F ";" ' NR==I {print $field }' $tempFile )

# gets data from the specific column and row

if [ -z "$tmp" ]

then

break

else

value=$(grep -n -w "$tmp" featureValues.txt | awk -F ":" '{print $1}') # get its value from the unique file

awk -v col="$fieldNum" -v val="$value" -v lNum="$i" -F ";" ' NR == lNum {$col=val; print}' $tempFile >> temp.txt

# change specific column in specific line and print it in temp file

sed -i '' 's/ /;/g' temp.txt # to get back the semicolun

fi

done

# remove unneded files

rm featureValuesTemp.txt

rm featureValues.txt

cp temp.txt $tempFile

rm temp.txt

dataSaved=0 # may be we will use this option after we save the data

else

clear

echo

echo "Something wrong in data feature name ($feature) :("

echo

echo "BACK TO MAIN MENU . . . ."

sleep 1

fi

fi

else

echo

echo "Please read the file first"

echo

echo Back to main menu ...

sleep 1

clear

fi

;;

o)

if [ $status -ne 0 ] # to check if the file is read or not

then

read -p " Please input the name of the categorical feature for label encoding : " feature # read the feature name from user

fieldNum=$( awk -v b="$feature" -F ";" ' NR==1 {for(i=1;i<=NF;i++){if($i ~ b){print i}}}' $tempFile ) #saves field num

if [ -z "$fieldNum" ] # to check if we found the feature or not

then # the feature doesn't exists

echo

echo "The name of categorical feature is wrong :("

echo

echo "BACK TO MAIN MENU . . . ."

echo

sleep 1

else

featureName=$(awk -v f=$fieldNum -F ";" ' NR==1 {print $f}' $tempFile ) # saves feature name from dataset

if [ "$feature" == "$featureName" ] # to check if the field have same feature name

then # its the same

(awk -v f=$fieldNum -F ";" ' NR>1 {print $f}' $tempFile > featureValuesTemp.txt )

# getting all feature inputs and add them to temp file ^^^

numOfEncodedStrings=$(wc -l featureValuesTemp.txt | awk -F " " '{ print $1 }')

# getting num of lines from featureValuesTemp.txt file ^^^

touch featureValues.txt # creating unique file to save feature inputs

for ((i=0; i <= $numOfEncodedStrings ; i++)) ## loop in temp file to check every data and add it to unique file

do

featureTemp=$(awk -v I=$i -F " " 'NR == I { print $1 }' featureValuesTemp.txt) # first row data saves

lineNum=$( grep -n -w "$featureTemp" featureValues.txt | awk -F ":" '{print $1}') #saves line num

if [ -z "$lineNum" ] # to check if we found the feature Value or not from unique file

then # when we cannot find the line num of it then the data is not added !!

echo $featureTemp >> featureValues.txt # then we will add it

lineNum=$( grep -n -w "$featureTemp" featureValues.txt | awk -F ":" '{print $1}') #saves field num

else

featureName=$(awk -v f=$lineNum -F " " ' NR==f {print $1}' featureValues.txt) # gets data from featureValues.txt (that conatins feature inputs)

fi

done

numOflinesInFeatureValues=$(wc -l featureValues.txt | awk -F " " '{print $1}') # gets num of lines in featureValues.txt

newString=""

for ((i=2; i <= $numOflinesInFeatureValues ; i++)) # started from i = 2 cause data starts from line 2

do

tmp=$(awk -v I=$i ' NR==I {print $1 }' featureValues.txt ) # get feature input from featureValues.txt (the unique file)

if [ -z "$tmp" ] # cause i used num of Encoded strings that have much data than featureValues.txt

then

break

else

newString+="$feature-$tmp:" # add the feature input to newString

fi

done

awk -F ";" -v F=$fieldNum -v V=$newString 'NR==1 {$F=V;print }' $tempFile > temp2.txt # change the value of feature in line i to the new value #new

sed -i '' '1s/:/ /g' temp2.txt # to remove the semicolun . '' for -i option in mac. 1s for first line and :/ /g for replace : with space

numOflinesInDataset=$(wc -l $tempFile | awk -F " " '{print $1}') # gets num of lines in dataset

numOflinesInDataset\_plusOne=$numOflinesInDataset

numOflinesInDataset\_plusOne=$(expr $numOflinesInDataset\_plusOne + 1) # to use last line

for ((i=2; i <= $numOflinesInDataset\_plusOne ; i++)) # started from i = 2 cause data starts from line 2 (line 1 is header)

do

newString=""

tmp2=""

for ((j=2; j <= $numOflinesInFeatureValues ; j++)) # started from i = 2 cause data starts from line 2

do

tmp=$(awk -v I=$j ' NR==I {print $1 }' featureValues.txt )

value=$(awk -F ";" -v I=$i -v F=$fieldNum ' NR==I {print $F }' $tempFile )

if [ "$tmp" == "$value" ] # to check if the data is the same as the feature value

then

newString+="1:" # if it is the same then we will add 1 to the new string. : to separate between values and remove it later

tmp2=$tmp

else

newString+="0:" # if it is not the same then we will add 0 to the new string. : to separate between values and remove it later

fi

done

awk -F ";" -v F=$fieldNum -v L=$i -v V=$newString 'NR==L {$F=V;print }' $tempFile >> temp2.txt # change the value of feature in line i to the new value

# change the value of feature in line i to the new value .'1' to tell awk to print the modified val

done

sed -i '' 's/:/ /g' temp2.txt # to remove the last char which is ':'

sed -i '' 's/ / /g' temp2.txt # to remove the last char which is ':'

sed -i '' 's/ /;/g' temp2.txt # to remove the last char which is ':'

cp temp2.txt $tempFile # to copy the new file to the temp file #new

rm featureValuesTemp.txt

rm featureValues.txt

dataSaved=0 # may be we will use this option after we save the data

echo

echo "DONE !"

echo

sleep 1

else

echo "Action canceled"

echo " Something wrong with feature name :)"

echo "BACK TO MAIN MENU . . . ."

echo

sleep 1

fi

fi

else

echo

echo Please read the file first

echo

sleep 1

echo Back to main menu ...

sleep 1

fi

;;

m)

if [ $status -ne 0 ] # to check if the file readed

then

read -p " Please input the name of the feature to be scaled : " feature # read the feature name from user

fieldNum=$( awk -v b="$feature" -F ";" 'NR==1 {for(i=1;i<=NF;i++){if($i ~ b ){print i}}}' $tempFile )

#saves field num, if the feature name is not found it will be empty

if [ -z "$fieldNum" ] # to check if we found the feature or not

then # the feature doesn't exists

echo "The name of categorical feature is wrong :("

else # found it maybe

value=$(awk -F ";" -v F=$fieldNum ' NR==2 {print $F }' $tempFile ) # gets the value of feature data to check if it is an integer or not

if [[ $value =~ ^[0-9]+$ ]]; then # '^' first character, '$' last character, '+' one or more times

# echo "Value is an integer"

awk -v col="$fieldNum" -F ";" 'NR>=2 {print $col }' $tempFile >> temp.txt

sort -n temp.txt > temp2.txt # sort the data in temp file and add it to temp2 file

rm temp.txt # remove temp file Useless

min=$(head -n 1 temp2.txt) # get the min value

max=$(tail -n 1 temp2.txt) # get the max value

echo

echo "min is $min"

echo "max is $max"

rm temp2.txt # remove temp2 file Useless

numlines=$(wc -l $tempFile | awk -F " " '{ print $1 }') # get the number of lines in dataset

touch temp.txt # create temp file

awk 'NR==1 {print }' $tempFile > temp.txt # add the first line to temp file

numlines=$(expr $numlines + 1) # get the number of lines in dataset

for ((i=2; i <= $numlines ; i++)) # loop in dataset to change the values

do

fieldVal=$(awk -F ";" -v F=$fieldNum -v L=$i ' NR==L {print $F }' $tempFile ) # get the value of feature in line i

a=$(echo "$fieldVal - $min" | bc) # get the value - min

b=$(echo "$max - $min" | bc) # get the max - min

newVal=$(echo "scale=3; $a / $b" | bc) # scale=3 means 3 decimal places after the decimal point

#bc is the command to calculate

newVal=$(printf "%.2f\n" $newVal) # print the result with 2 decimal places

awk -v F=$fieldNum -v L=$i -v V=$newVal -F ";" 'NR==L {$F=V}1' $tempFile > temp2.txt # change the value of feature in line i to the new value

cp temp2.txt $tempFile # copy the temp2 file to dataset file

sed -i '' 's/ /;/g' $tempFile # replace " " with ; in dataset file to make it as input file ,, the '' cause im using mac

done

rm temp.txt # remove temp file Useless

rm temp2.txt # remove temp2 file Useless

dataSaved=0 # may be we will use this option after we save the data

else

echo this feature is categorical feature and must be encoded first

sleep 1

echo Back to main menu ...

sleep 1

fi

fi

else

echo Please read the file first

sleep 1

echo Back to main menu ...

sleep 1

fi

;;

s)

if [ $status -ne 0 ] # to check if the file readed

then

read -p " Please input the name of the file to save the processed dataset (do not include .txt) : " newFile # read the feature name from user

newFile=$newFile.txt

cp $tempFile $newFile # copy the temp file to the new file

echo

echo "The processed dataset saved in $newFile"

dataSaved=1 # to check if the data saved or not

echo

sleep 1

echo Back to main menu ...

echo

echo

sleep 1

else

echo "There is no date to be saved ( No File Readed )"

sleep 1

echo Back to main menu ...

sleep 1

fi

;;

e)

if [ $dataSaved -eq 0 ] # to check if the file saved

then

read -p " The processed dataset is not saved. Are you sure you want to exit? " answer

if [ $answer == "yes" ] || [ $answer == "y" ] || [ $answer == "Y" ] || [ $answer == "YES" ] || [ $answer == "Yes" ]

then

echo

echo Exiting ...

sleep 1

exit 1

else

echo

echo Back to main menu ...

sleep 1

fi

else

echo

echo "The processed dataset is saved"

echo

read -p " Are you sure you want to exit? " answer

if [ $answer == "yes" ] || [ $answer == "y" ] || [ $answer == "Y" ] || [ $answer == "YES" ] || [ $answer == "Yes" ]

then

echo

echo Exiting ...

sleep 1

exit 1

else

echo

echo Back to main menu ...

sleep 1

fi

fi

;;

\*)

clear

echo " thats not an option"

sleep 2

;;

esac

done

# after checked when just press enter that may cause error

#(not mentioned to check it)

**Idea :**

Build a shell script program that show the user a list of options and ask the user

to choose . if he chose ‘r’ then the program should ask him to type the name of file . and each encoder option should ask for the name of the feature to be encoded. and check the existence of each file and feature added from user.

r)

code :

read -p "Please input the name of the dataset file (do not include .txt) : " fileName # read the file name from user

fileName=$fileName.txt

if [ -e $fileName ] # checks if file exists

then

fs1=$( awk -F ";" 'NR ==1 {print NF }' $fileName ) # get num of data fields

fs2=$( awk -F ";" 'NR ==2 {print NF }' $fileName ) # get num of data fields

if [ $fs1 -eq $fs2 ] # to check if the file is clean or not

then

echo

clear

echo "file is read successfully :)"

echo

echo "------------------------------ FILE Features ------------------------------"

fs=$(awk -F ";" 'NR==1{print $0}' $fileName | sed 's/;/ /g')

echo $fs

echo "-------------------------------- DONE ---------------------------------"

cp $fileName $tempFile # making temporary dataset file to work on

status=1

else

echo "file is not clean"

fi

else

clear

echo "file with name ($fileName) does not exist :( "

echo

echo

fi

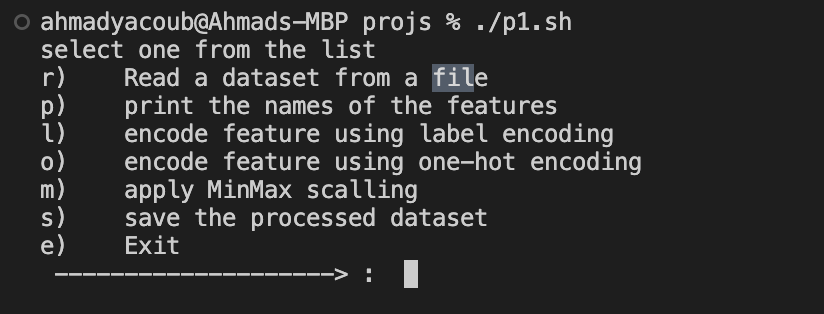
echo Back to main menu ...

echo

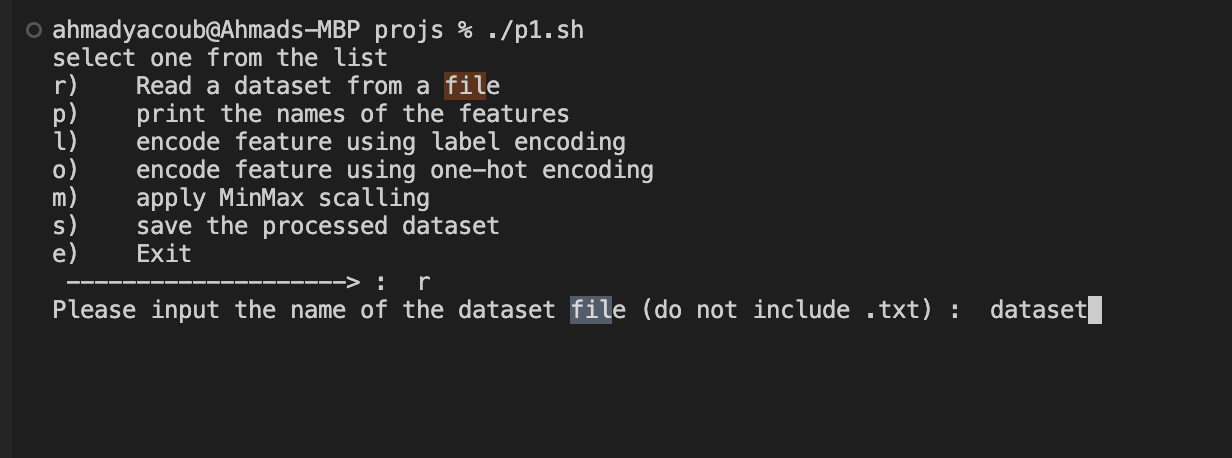
echo

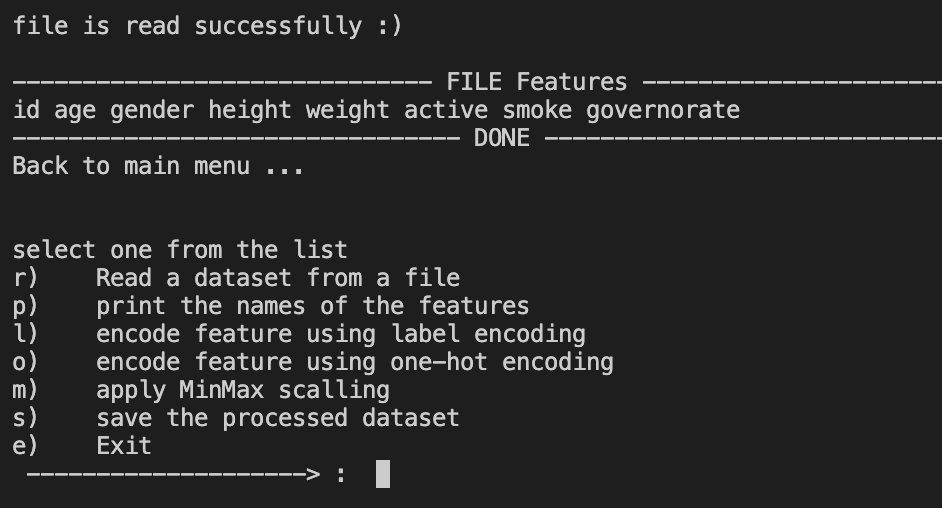
sleep 2

;;

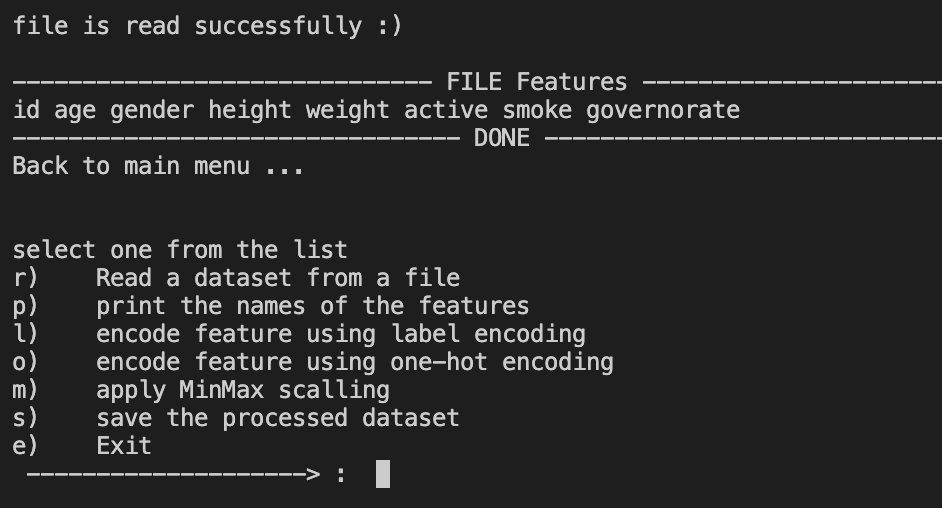


r.a=

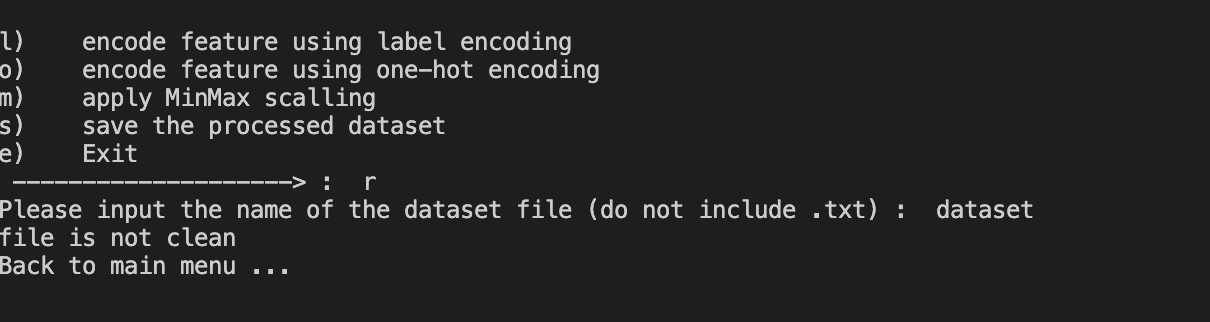




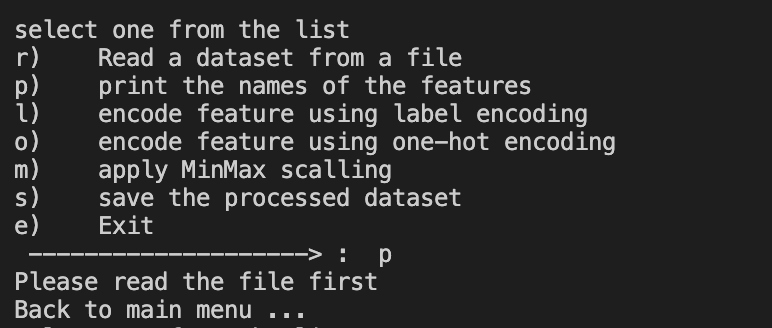
r.b : when add name of none existed file



r.c :



r.d :



p)

code :

p)

if [ $status -ne 0 ] # to check if the file readed

then

clear

echo

echo "------------------------------ FILE DATA ---------------------------"

echo

awk -F ";" '{$1=""; print $0}' $tempFile # print the data in the file

echo

echo "-------------------------------- DONE ------------------------------"

echo

sleep 2

else

echo "Please read the file first"

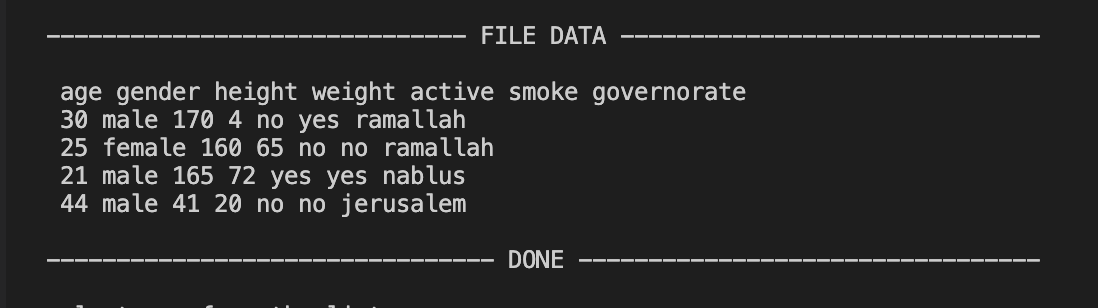
sleep 1

echo Back to main menu ...

sleep 1

fi

;;



P ) Output

l)

Code :

l)

if [ $status -ne 0 ] # to check if the file readed

then

read -p " Please input the name of the categorical feature for label encoding : " feature

# read the feature name from user - p to read from same line

fieldNum=$( awk -v b="$feature" -F ";" 'NR==1 {for(i=1;i<=NF;i++){if($i ~ b ){print i}}}' $tempFile) #saves field num

# in awk -v --> letting me to use variables in awk cuz of '$var'

if [ -z "$fieldNum" ] # to check if we found the feature or not

then # the feature doesn't exists

clear

echo

echo "The name of categorical feature ($feature) is wrong :("

echo

echo Back to main menu ...

sleep 2

else # found it maybe

featureName=$(awk -v f=$fieldNum -F ";" ' NR==1 {print $f}' $tempFile ) # saves feature name from dataset

if [ "$feature" == "$featureName" ] # to check if the field have same feature name

then # its the same

(awk -v f=$fieldNum -F ";" ' NR>1 {print $f}' $tempFile > featureValuesTemp.txt )

#NR>1 --> to skip the first line cuz it contains the feature name

# getting all feature inputs and add them to temp file to work on with

numOfEncodedStrings=$(wc -l featureValuesTemp.txt | awk -F " " '{ print $1 }')

# getting num of lines from featureValuesTemp.txt file ^^^

for ((i=0; i <= $numOfEncodedStrings ; i++))

# loop in temp file to check every feature input and add it to unique file if not added yet

do

featureTemp=$(awk -v I=$i -F " " 'NR == I { print $1 }' featureValuesTemp.txt) # first column data saves

# -F is unneciary cuz we have only one column in the file

touch featureValues.txt # to create the unique file of feature inputs

lineNum=$( grep -n -w "$featureTemp" featureValues.txt | awk -F ":" '{print $1}') #saves line num

# -n to print line number

if [ -z "$lineNum" ] # to check if we found the feature Value or not from unique file

then # when we cannot find the line num of it then the data is not added !!

echo $featureTemp >> featureValues.txt # then we will add it

lineNum=$( grep -n -w "$featureTemp" featureValues.txt | awk -F ":" '{print $1}') #saves field num

else

featureName=$(awk -v f=$lineNum -F " " ' NR==f {print $1}' featureValues.txt)

# gets data from featureValues.txt (that conatins feature inputs)

fi

done

echo

echo --------------------------------- LABEL ENCODING ---------------------------------

for ((i=2; i <= $numOfEncodedStrings ; i++)) # loop to print the feature inputs and their values

do

tmp=$(awk -v I=$i ' NR==I {print $1 }' featureValues.txt )

if [ -z "$tmp" ] # to check if we found the feature Value or not from unique file

then # when we cannot find the line num of it then the data is not added !!

break

else

echo "feature $tmp Value is : $i"

fi

done

echo "--------------------------------- feature Inputs ^^^ ---------------------------------"

numOfEncodedStrings=$(wc -l $tempFile | awk -F " " '{print $1}')

numOfEncodedStrings+=1 # to replace the last line

touch temp.txt

awk -F " " ' NR == 1 {print $1 }' $tempFile > temp.txt # to add the header to the temp file

for ((i=2; i <= $numOfEncodedStrings ; i++))

do

tmp=$(awk -v I=$i -v field=$fieldNum -F ";" ' NR==I {print $field }' $tempFile )

# gets data from the specific column and row

if [ -z "$tmp" ]

then

break

else

value=$(grep -n -w "$tmp" featureValues.txt | awk -F ":" '{print $1}') # get its value from the unique file

awk -v col="$fieldNum" -v val="$value" -v lNum="$i" -F ";" ' NR == lNum {$col=val; print}' $tempFile >> temp.txt

# change specific column in specific line and print it in temp file

sed -i '' 's/ /;/g' temp.txt # to get back the semicolun

fi

done

# remove unneded files

rm featureValuesTemp.txt

rm featureValues.txt

cp temp.txt $tempFile

rm temp.txt

dataSaved=0 # may be we will use this option after we save the data

else

clear

echo

echo "Something wrong in data feature name ($feature) :("

echo

echo "BACK TO MAIN MENU . . . ."

sleep 1

fi

fi

else

echo

echo "Please read the file first"

echo

echo Back to main menu ...

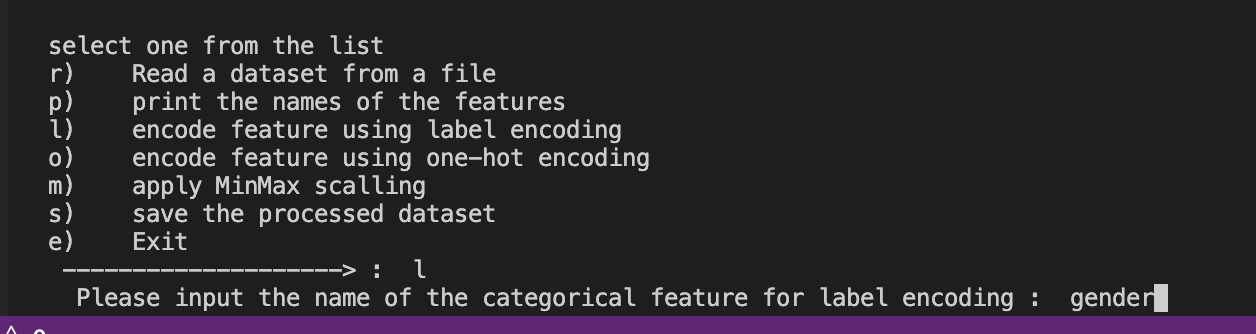
sleep 1

clear

fi

;;

l.a :



l.b :

when enters feature not exists



l.c :



O )

Code :

if [ $status -ne 0 ] # to check if the file is read or not

then

read -p " Please input the name of the categorical feature for label encoding : " feature # read the feature name from user

fieldNum=$( awk -v b="$feature" -F ";" ' NR==1 {for(i=1;i<=NF;i++){if($i ~ b){print i}}}' $tempFile ) #saves field num

if [ -z "$fieldNum" ] # to check if we found the feature or not

then # the feature doesn't exists

echo

echo "The name of categorical feature is wrong :("

echo

echo "BACK TO MAIN MENU . . . ."

echo

sleep 1

else

featureName=$(awk -v f=$fieldNum -F ";" ' NR==1 {print $f}' $tempFile ) # saves feature name from dataset

if [ "$feature" == "$featureName" ] # to check if the field have same feature name

then # its the same

(awk -v f=$fieldNum -F ";" ' NR>1 {print $f}' $tempFile > featureValuesTemp.txt )

# getting all feature inputs and add them to temp file ^^^

numOfEncodedStrings=$(wc -l featureValuesTemp.txt | awk -F " " '{ print $1 }')

# getting num of lines from featureValuesTemp.txt file ^^^

touch featureValues.txt # creating unique file to save feature inputs

for ((i=0; i <= $numOfEncodedStrings ; i++)) ## loop in temp file to check every data and add it to unique file

do

featureTemp=$(awk -v I=$i -F " " 'NR == I { print $1 }' featureValuesTemp.txt) # first row data saves

lineNum=$( grep -n -w "$featureTemp" featureValues.txt | awk -F ":" '{print $1}') #saves line num

if [ -z "$lineNum" ] # to check if we found the feature Value or not from unique file

then # when we cannot find the line num of it then the data is not added !!

echo $featureTemp >> featureValues.txt # then we will add it

lineNum=$( grep -n -w "$featureTemp" featureValues.txt | awk -F ":" '{print $1}') #saves field num

else

featureName=$(awk -v f=$lineNum -F " " ' NR==f {print $1}' featureValues.txt) # gets data from featureValues.txt (that conatins feature inputs)

fi

done

numOflinesInFeatureValues=$(wc -l featureValues.txt | awk -F " " '{print $1}') # gets num of lines in featureValues.txt

newString=""

for ((i=2; i <= $numOflinesInFeatureValues ; i++)) # started from i = 2 cause data starts from line 2

do

tmp=$(awk -v I=$i ' NR==I {print $1 }' featureValues.txt ) # get feature input from featureValues.txt (the unique file)

if [ -z "$tmp" ] # cause i used num of Encoded strings that have much data than featureValues.txt

then

break

else

newString+="$feature-$tmp:" # add the feature input to newString

fi

done

awk -F ";" -v F=$fieldNum -v V=$newString 'NR==1 {$F=V;print }' $tempFile > temp2.txt # change the value of feature in line i to the new value #new

sed -i '' '1s/:/ /g' temp2.txt # to remove the semicolun . '' for -i option in mac. 1s for first line and :/ /g for replace : with space

numOflinesInDataset=$(wc -l $tempFile | awk -F " " '{print $1}') # gets num of lines in dataset

numOflinesInDataset\_plusOne=$numOflinesInDataset

numOflinesInDataset\_plusOne=$(expr $numOflinesInDataset\_plusOne + 1) # to use last line

for ((i=2; i <= $numOflinesInDataset\_plusOne ; i++)) # started from i = 2 cause data starts from line 2 (line 1 is header)

do

newString=""

tmp2=""

for ((j=2; j <= $numOflinesInFeatureValues ; j++)) # started from i = 2 cause data starts from line 2

do

tmp=$(awk -v I=$j ' NR==I {print $1 }' featureValues.txt )

value=$(awk -F ";" -v I=$i -v F=$fieldNum ' NR==I {print $F }' $tempFile )

if [ "$tmp" == "$value" ] # to check if the data is the same as the feature value

then

newString+="1:" # if it is the same then we will add 1 to the new string. : to separate between values and remove it later

tmp2=$tmp

else

newString+="0:" # if it is not the same then we will add 0 to the new string. : to separate between values and remove it later

fi

done

awk -F ";" -v F=$fieldNum -v L=$i -v V=$newString 'NR==L {$F=V;print }' $tempFile >> temp2.txt # change the value of feature in line i to the new value

# change the value of feature in line i to the new value .'1' to tell awk to print the modified val

done

sed -i '' 's/:/ /g' temp2.txt # to remove the last char which is ':'

sed -i '' 's/ / /g' temp2.txt # to remove the last char which is ':'

sed -i '' 's/ /;/g' temp2.txt # to remove the last char which is ':'

cp temp2.txt $tempFile # to copy the new file to the temp file #new

rm featureValuesTemp.txt

rm featureValues.txt

dataSaved=0 # may be we will use this option after we save the data

echo

echo "DONE !"

echo

sleep 1

else

echo "Action canceled"

echo " Something wrong with feature name :)"

echo "BACK TO MAIN MENU . . . ."

echo

sleep 1

fi

fi

else

echo

echo Please read the file first

echo

sleep 1

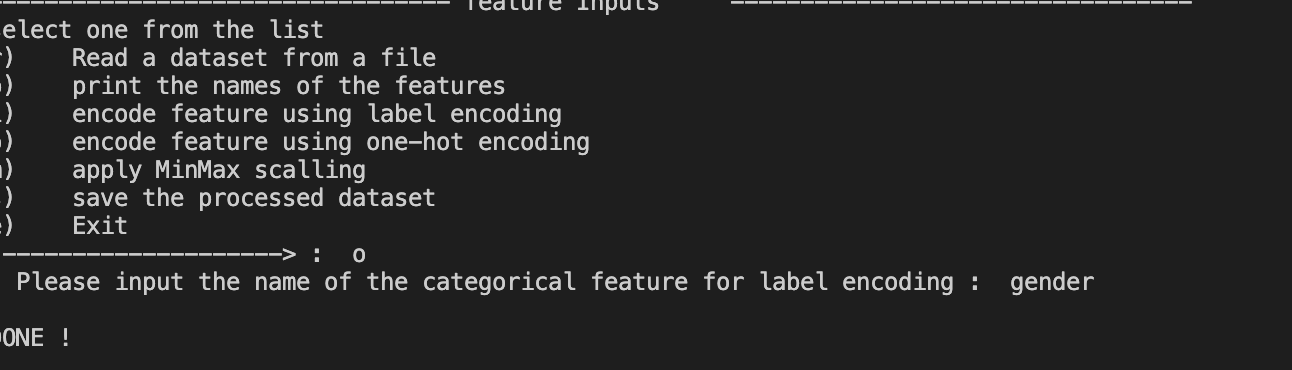
echo Back to main menu ...

sleep 1

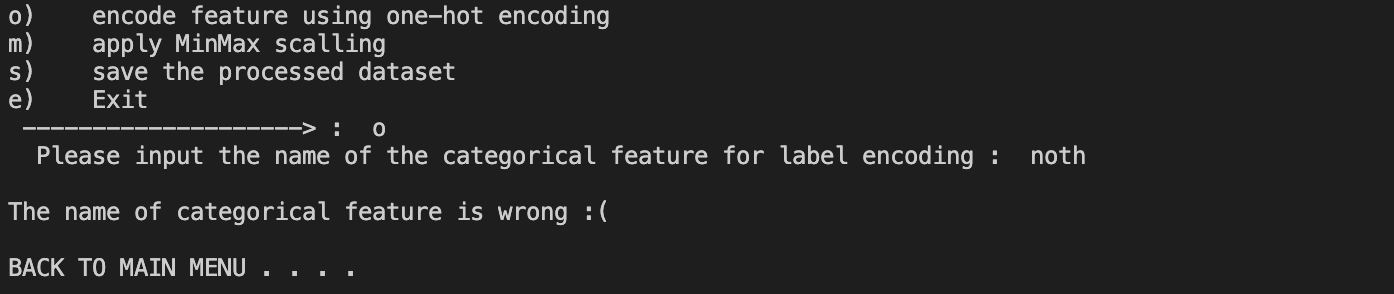
fi

;;

o.a:



o.b :



m) :

code :

m)

if [ $status -ne 0 ] # to check if the file readed

then

read -p " Please input the name of the feature to be scaled : " feature # read the feature name from user

fieldNum=$( awk -v b="$feature" -F ";" 'NR==1 {for(i=1;i<=NF;i++){if($i ~ b ){print i}}}' $tempFile )

#saves field num, if the feature name is not found it will be empty

if [ -z "$fieldNum" ] # to check if we found the feature or not

then # the feature doesn't exists

echo "The name of categorical feature is wrong :("

else # found it maybe

value=$(awk -F ";" -v F=$fieldNum ' NR==2 {print $F }' $tempFile ) # gets the value of feature data to check if it is an integer or not

if [[ $value =~ ^[0-9]+$ ]]; then # '^' first character, '$' last character, '+' one or more times

# echo "Value is an integer"

awk -v col="$fieldNum" -F ";" 'NR>=2 {print $col }' $tempFile >> temp.txt

sort -n temp.txt > temp2.txt # sort the data in temp file and add it to temp2 file

rm temp.txt # remove temp file Useless

min=$(head -n 1 temp2.txt) # get the min value

max=$(tail -n 1 temp2.txt) # get the max value

echo

echo "min is $min"

echo "max is $max"

rm temp2.txt # remove temp2 file Useless

numlines=$(wc -l $tempFile | awk -F " " '{ print $1 }') # get the number of lines in dataset

touch temp.txt # create temp file

awk 'NR==1 {print }' $tempFile > temp.txt # add the first line to temp file

numlines=$(expr $numlines + 1) # get the number of lines in dataset

for ((i=2; i <= $numlines ; i++)) # loop in dataset to change the values

do

fieldVal=$(awk -F ";" -v F=$fieldNum -v L=$i ' NR==L {print $F }' $tempFile ) # get the value of feature in line i

a=$(echo "$fieldVal - $min" | bc) # get the value - min

b=$(echo "$max - $min" | bc) # get the max - min

newVal=$(echo "scale=3; $a / $b" | bc) # scale=3 means 3 decimal places after the decimal point

#bc is the command to calculate

newVal=$(printf "%.2f\n" $newVal) # print the result with 2 decimal places

awk -v F=$fieldNum -v L=$i -v V=$newVal -F ";" 'NR==L {$F=V}1' $tempFile > temp2.txt # change the value of feature in line i to the new value

cp temp2.txt $tempFile # copy the temp2 file to dataset file

sed -i '' 's/ /;/g' $tempFile # replace " " with ; in dataset file to make it as input file ,, the '' cause im using mac

done

rm temp.txt # remove temp file Useless

rm temp2.txt # remove temp2 file Useless

dataSaved=0 # may be we will use this option after we save the data

else

echo this feature is categorical feature and must be encoded first

sleep 1

echo Back to main menu ...

sleep 1

fi

fi

else

echo Please read the file first

sleep 1

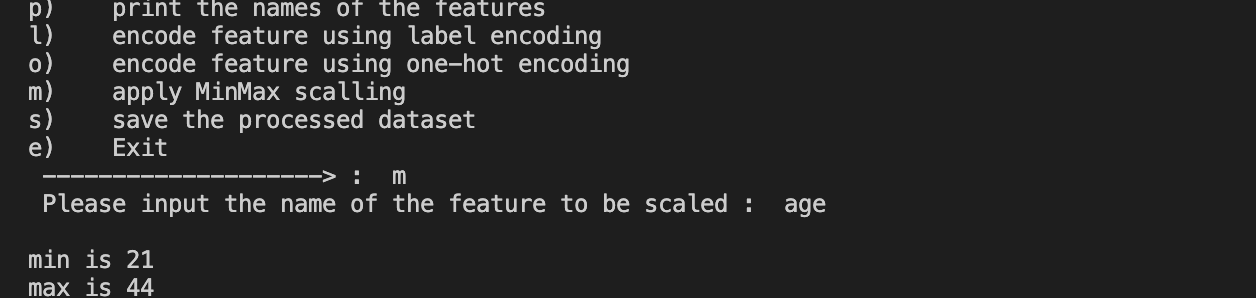
echo Back to main menu ...

sleep 1

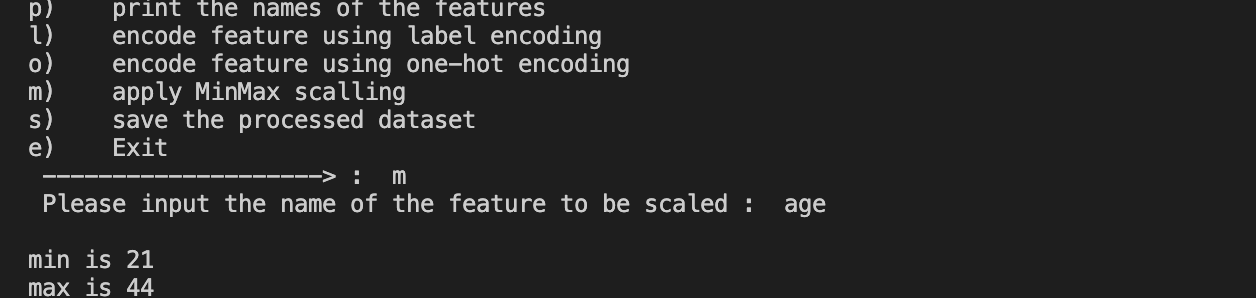
fi

;;

m.a , m.c:



m.b:



s)

code:

if [ $status -ne 0 ] # to check if the file readed

then

read -p " Please input the name of the file to save the processed dataset (do not include .txt) : " newFile # read the feature name from user

newFile=$newFile.txt

cp $tempFile $newFile # copy the temp file to the new file

echo

echo "The processed dataset saved in $newFile"

dataSaved=1 # to check if the data saved or not

echo

sleep 1

echo Back to main menu ...

echo

echo

sleep 1

else

echo "There is no date to be saved ( No File Readed )"

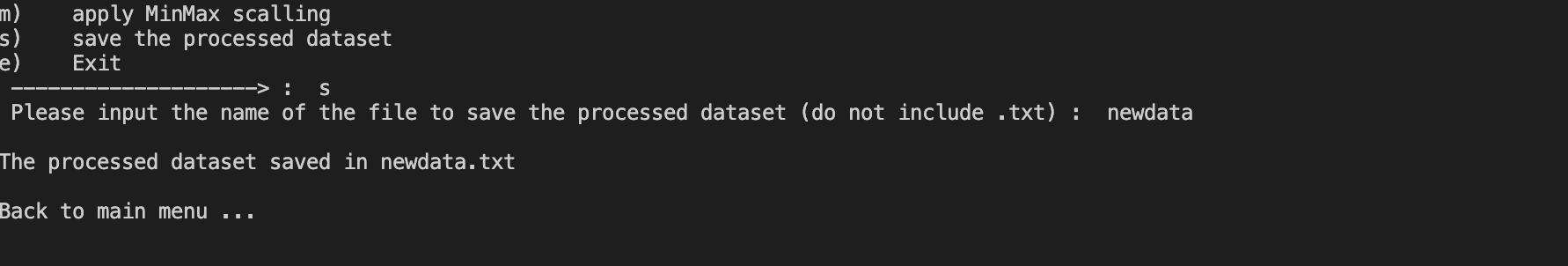
sleep 1

echo Back to main menu ...

sleep 1

fi

;;

--

e)

code

if [ $dataSaved -eq 0 ] # to check if the file saved

then

read -p " The processed dataset is not saved. Are you sure you want to exit? " answer

if [ $answer == "yes" ] || [ $answer == "y" ] || [ $answer == "Y" ] || [ $answer == "YES" ] || [ $answer == "Yes" ]

then

echo

echo Exiting ...

sleep 1

exit 1

else

echo

echo Back to main menu ...

sleep 1

fi

else

echo

echo "The processed dataset is saved"

echo

read -p " Are you sure you want to exit? " answer

if [ $answer == "yes" ] || [ $answer == "y" ] || [ $answer == "Y" ] || [ $answer == "YES" ] || [ $answer == "Yes" ]

then

echo

echo Exiting ...

sleep 1

exit 1

else

echo

echo Back to main menu ...

sleep 1

fi

fi

;;

e.a:

