جامعة بيرزيت

BIRZEIT UNIVERSITY

Department of Electrical & Computer Engineering

First Semester, 2022/2023

ENCS3130 Linux Laboratory

Shell Scripting Project – Data Preprocessing

Prepared By:

Omar Masalmah "1200060"

Instructor :

D. Mohammad Jubran

Section : 2

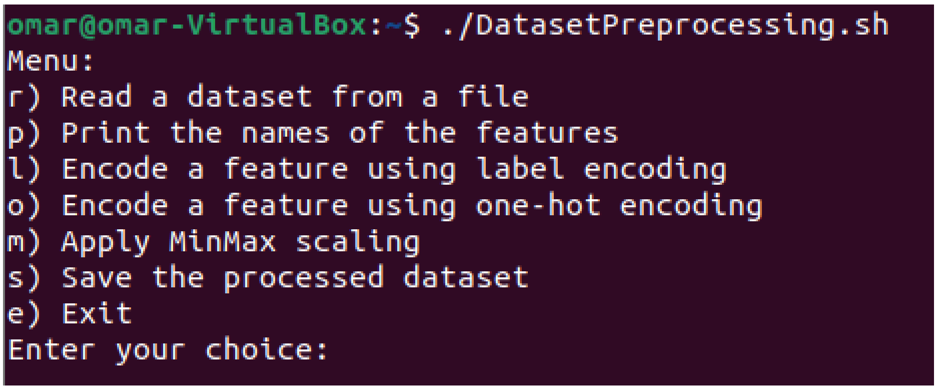
Date : 1/1/2023

The idea of project:

The project involves creating a shell script that provides various options for processing a dataset read from a file. The options include reading a dataset from a file, printing the names of the features, encoding a feature using label encoding or one-hot encoding, applying MinMax scaling, saving the processed dataset, and exiting the program.

The script should handle a number of different scenarios, such as verifying that the file specified by the user exists before reading it, checking the format of the data in the file, and verifying that a dataset has been read from a file before attempting to perform any other actions on it. Additionally, the script should handle errors, such as when the user enters an invalid option or specifies a feature that does not exist in the dataset.

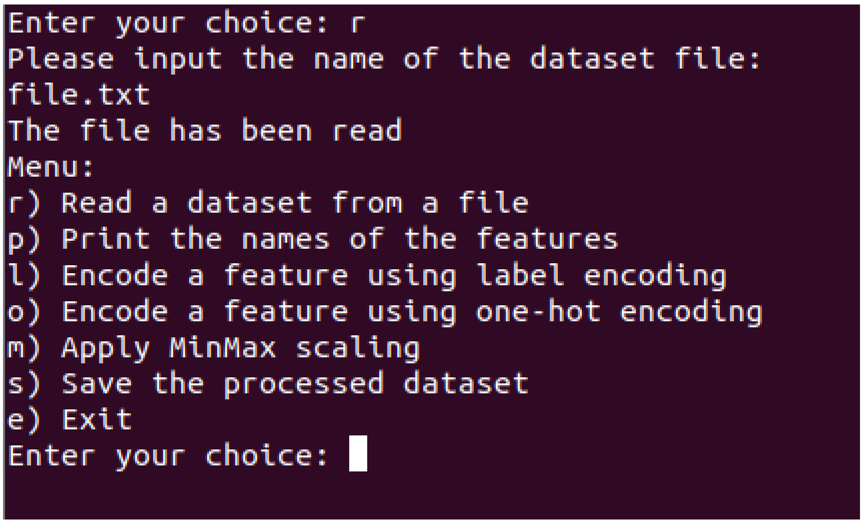
Screenshots:



```
omar@omar-VirtualBox:~$ ./DatasetPreprocessing.sh
Menu:
r) Read a dataset from a file
p) Print the names of the features
l) Encode a feature using label encoding
o) Encode a feature using one-hot encoding
m) Apply MinMax scaling
s) Save the processed dataset
e) Exit
Enter your choice:
```

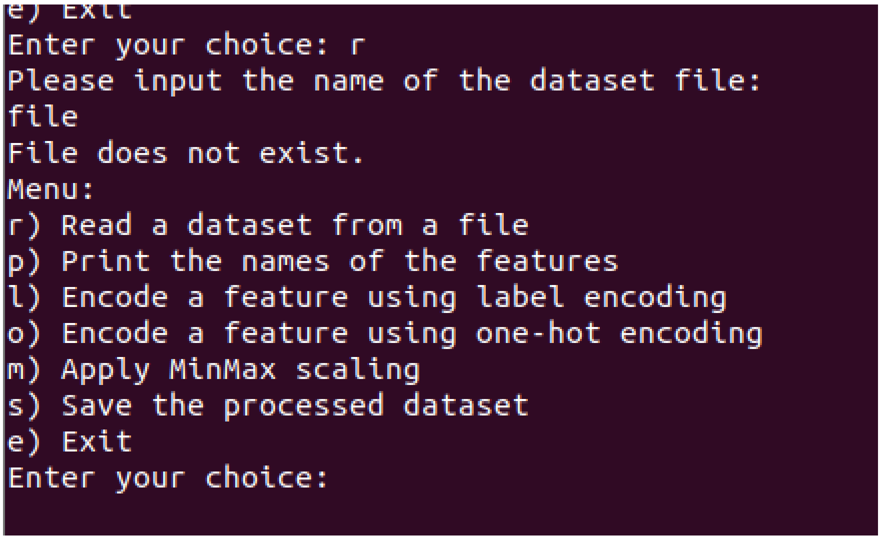
Here is the menu when you start the project.

Read file option



```
Enter your choice: r
Please input the name of the dataset file:
file.txt
The file has been read
Menu:
r) Read a dataset from a file
p) Print the names of the features
l) Encode a feature using label encoding
o) Encode a feature using one-hot encoding
m) Apply MinMax scaling
s) Save the processed dataset
e) Exit
Enter your choice: █
```

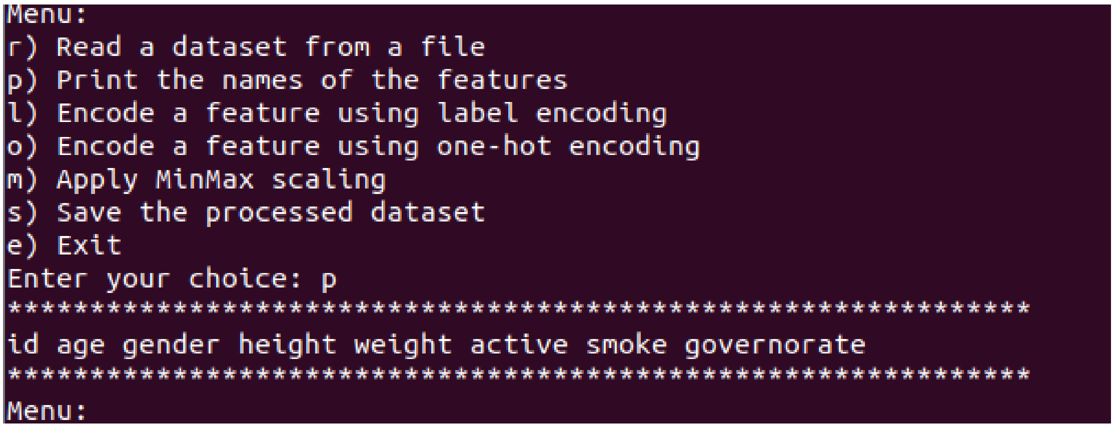
First option is read the file, if you entered an exist file it will read it and print an acceptance message.



```
e) Exit
Enter your choice: r
Please input the name of the dataset file:
file
File does not exist.
Menu:
r) Read a dataset from a file
p) Print the names of the features
l) Encode a feature using label encoding
o) Encode a feature using one-hot encoding
m) Apply MinMax scaling
s) Save the processed dataset
e) Exit
Enter your choice:
```

If you entered a file name and it not found.

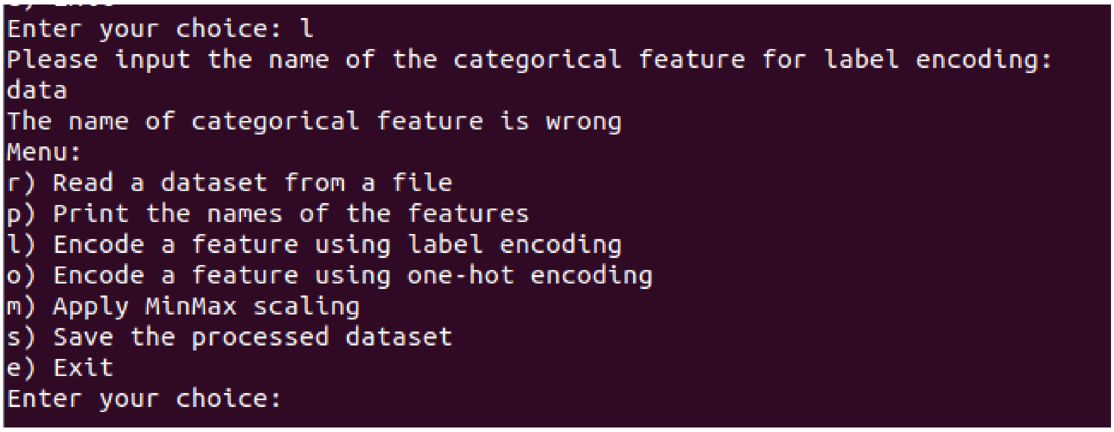
Print the names of feature option



```
Menu:
r) Read a dataset from a file
p) Print the names of the features
l) Encode a feature using label encoding
o) Encode a feature using one-hot encoding
m) Apply MinMax scaling
s) Save the processed dataset
e) Exit
Enter your choice: p
**************************************************************
id age gender height weight active smoke governorate
**************************************************************
Menu:
```

If you enter p letter it will Print the names of feature option.

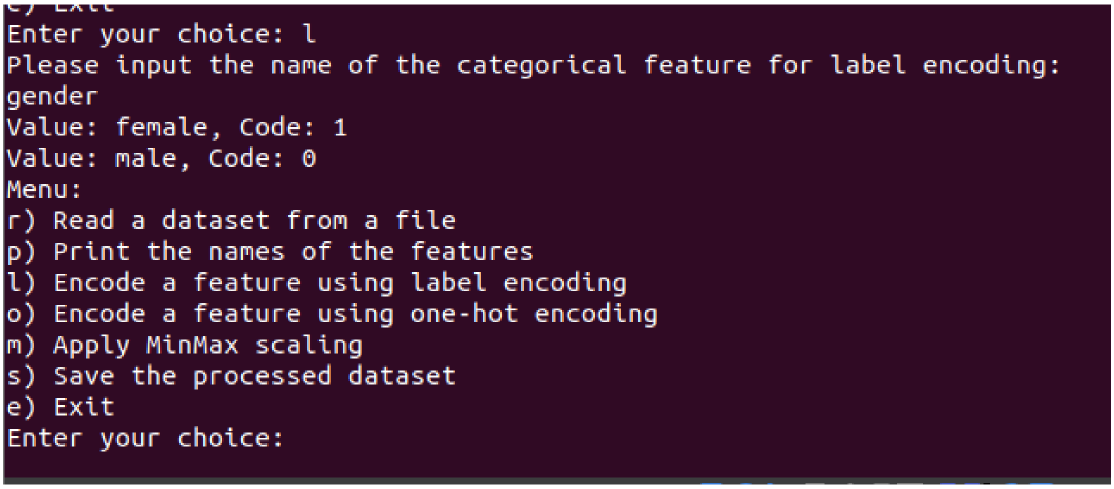
Label encoding option



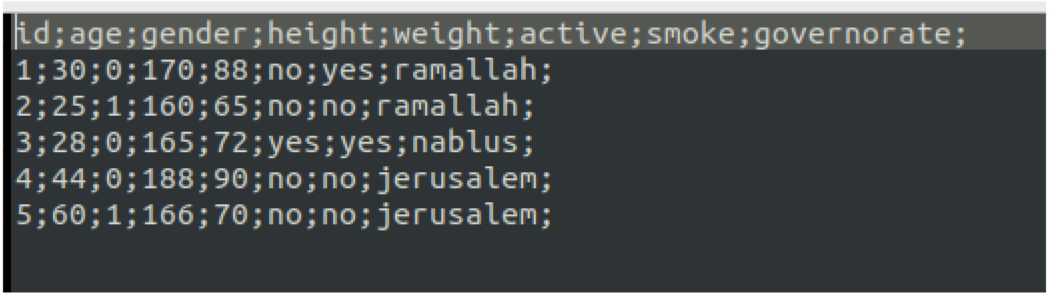
```
Enter your choice: l
Please input the name of the categorical feature for label encoding:
data
The name of categorical feature is wrong
Menu:
r) Read a dataset from a file
p) Print the names of the features
l) Encode a feature using label encoding
o) Encode a feature using one-hot encoding
m) Apply MinMax scaling
s) Save the processed dataset
e) Exit
Enter your choice:
```

When you chose that option it will ask you to enter the categorical feature name, then it will check if it exist, in that photo it's not exist.

Here the feature are exist and it will give each value a code.



```
Enter your choice: l
Please input the name of the categorical feature for label encoding:
gender
Value: female, Code: 1
Value: male, Code: 0
Menu:
r) Read a dataset from a file
p) Print the names of the features
l) Encode a feature using label encoding
o) Encode a feature using one-hot encoding
m) Apply MinMax scaling
s) Save the processed dataset
e) Exit
Enter your choice:
```

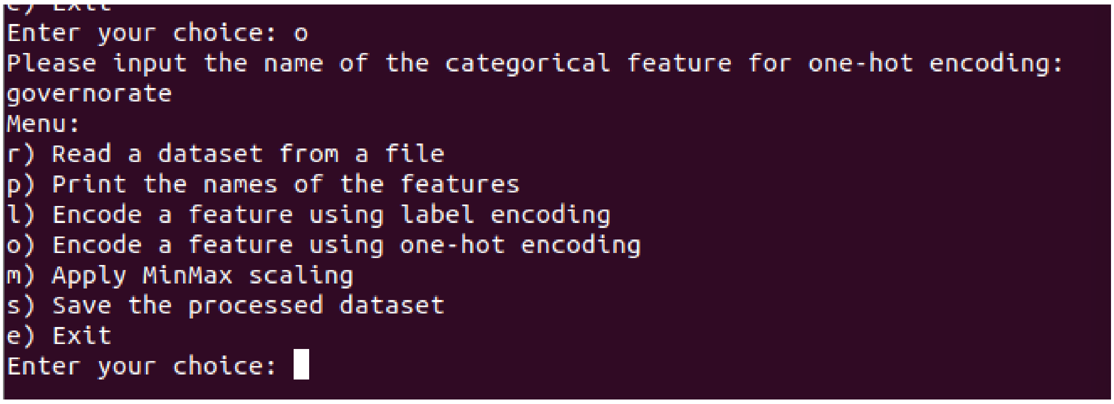


```
id;age;gender;height;weight;active;smoke;governorate;
1;30;0;170;88;no;yes;ramallah;
2;25;1;160;65;no;no;ramallah;
3;28;0;165;72;yes;yes;nablus;
4;44;0;188;90;no;no;jerusalem;
5;60;1;166;70;no;no;jerusalem;
```

Here the output of previous step.

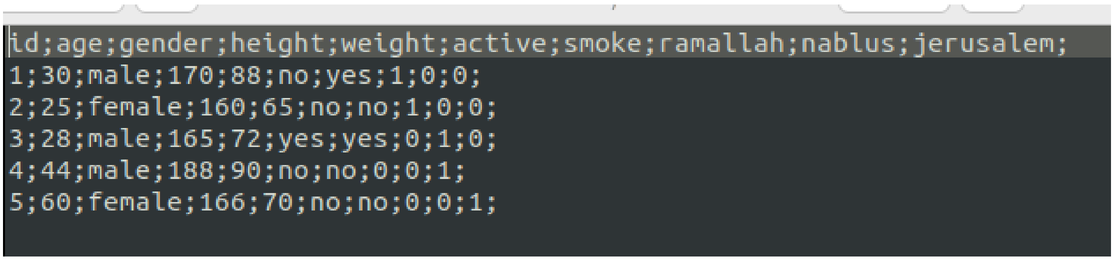
One-Hot encoding option

As label encoding it will ask for the name of feature and check if it exsist or not.



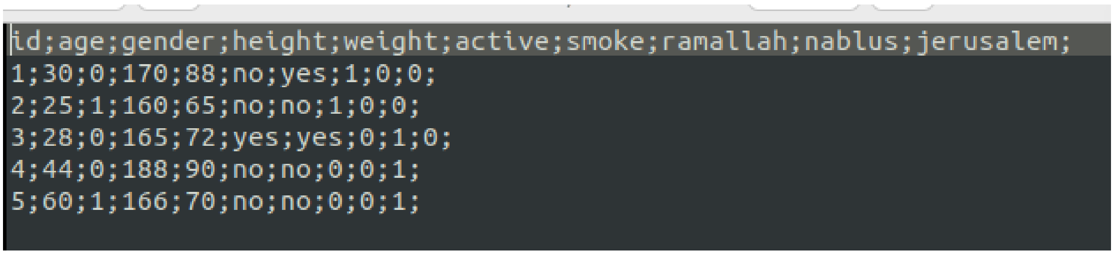
```
e) Exit
Enter your choice: o
Please input the name of the categorical feature for one-hot encoding:
governorate
Menu:
r) Read a dataset from a file
p) Print the names of the features
l) Encode a feature using label encoding
o) Encode a feature using one-hot encoding
m) Apply MinMax scaling
s) Save the processed dataset
e) Exit
Enter your choice: 
```

Here the output of one-hot encoding for governorate feature.



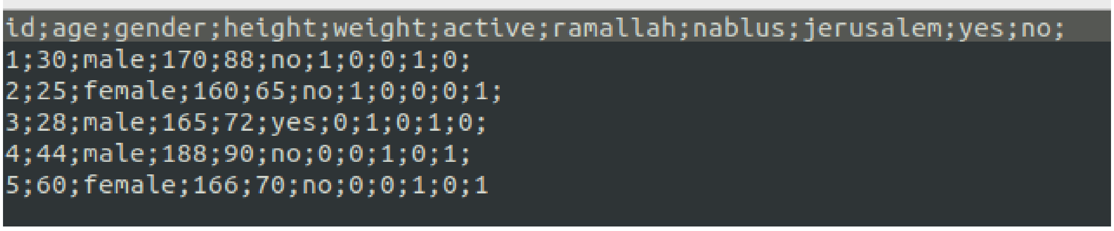
```
id;age;gender;height;weight;active;smoke;ramallah;nablus;jerusalem;
1;30;male;170;88;no;yes;1;0;0;
2;25;female;160;65;no;no;1;0;0;
3;28;male;165;72;yes;yes;0;1;0;
4;44;male;188;90;no;no;0;0;1;
5;60;female;166;70;no;no;0;0;1;
```

Here the output if we choose label option for gender and then one-hot option for governorate feature.



```
id;age;gender;height;weight;active;smoke;ramallah;nablus;jerusalem;
1;30;0;170;88;no;yes;1;0;0;
2;25;1;160;65;no;no;1;0;0;
3;28;0;165;72;yes;yes;0;1;0;
4;44;0;188;90;no;no;0;0;1;
5;60;1;166;70;no;no;0;0;1;
```

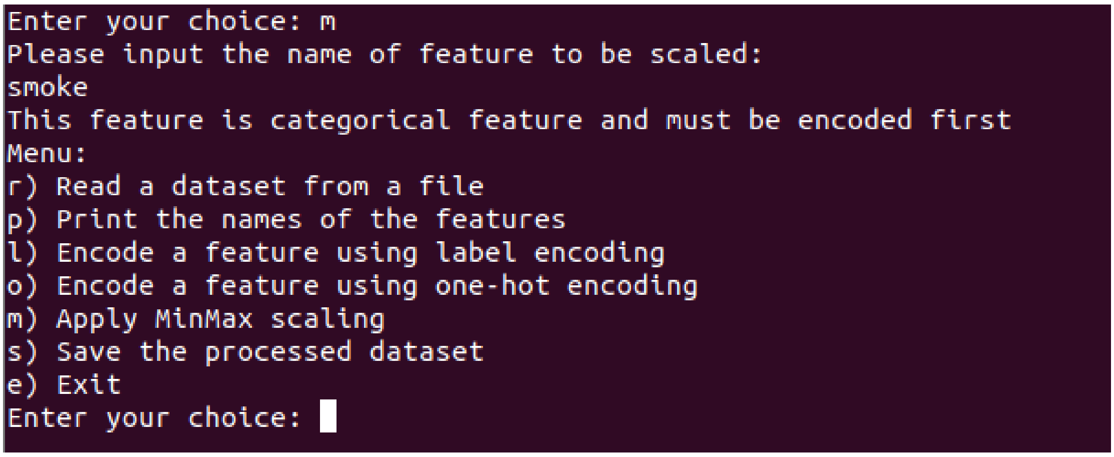
If we choose one-hot option twice for governorate and smoke the output will be.



```
id;age;gender;height;weight;active;ramallah;nablus;jerusalem;yes;no;
1;30;male;170;88;no;1;0;0;1;0;
2;25;female;160;65;no;1;0;0;0;1;
3;28;male;165;72;yes;0;1;0;1;0;
4;44;male;188;90;no;0;0;1;0;1;
5;60;female;166;70;no;0;0;1;0;1
```

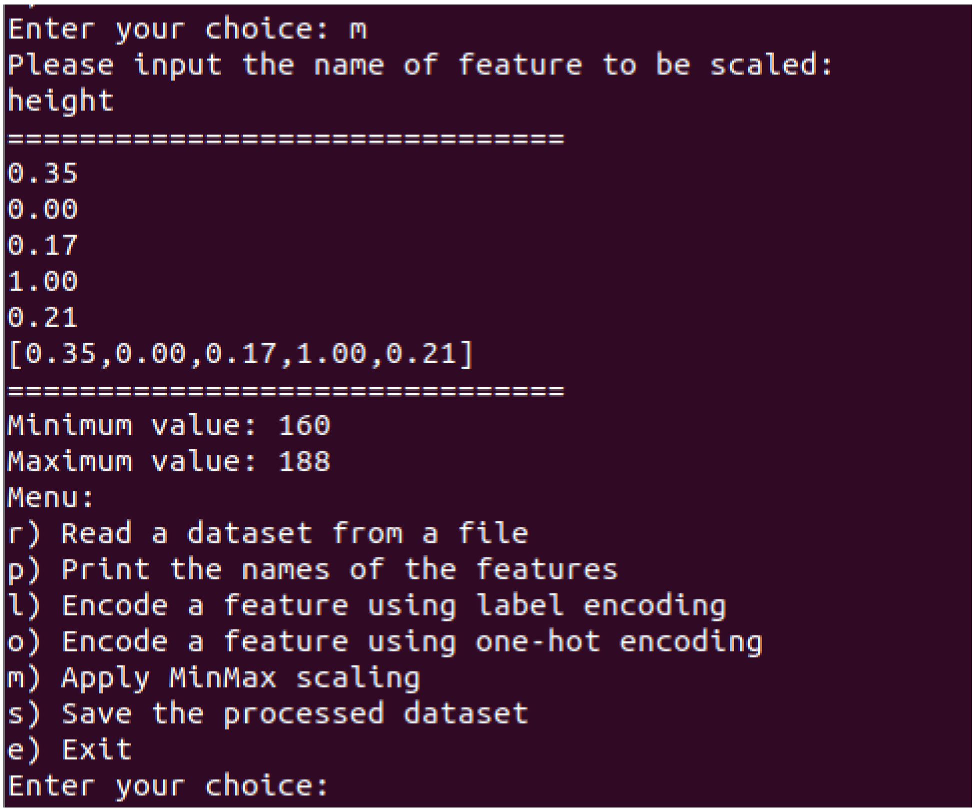
MinMax scaling option

MinMax scaling is a method used to transform the values of a feature in a dataset so that they are between a given minimum and maximum value, typically 0 and 1.



```
Enter your choice: m
Please input the name of feature to be scaled:
smoke
This feature is categorical feature and must be encoded first
Menu:
r) Read a dataset from a file
p) Print the names of the features
l) Encode a feature using label encoding
o) Encode a feature using one-hot encoding
m) Apply MinMax scaling
s) Save the processed dataset
e) Exit
Enter your choice: 
```

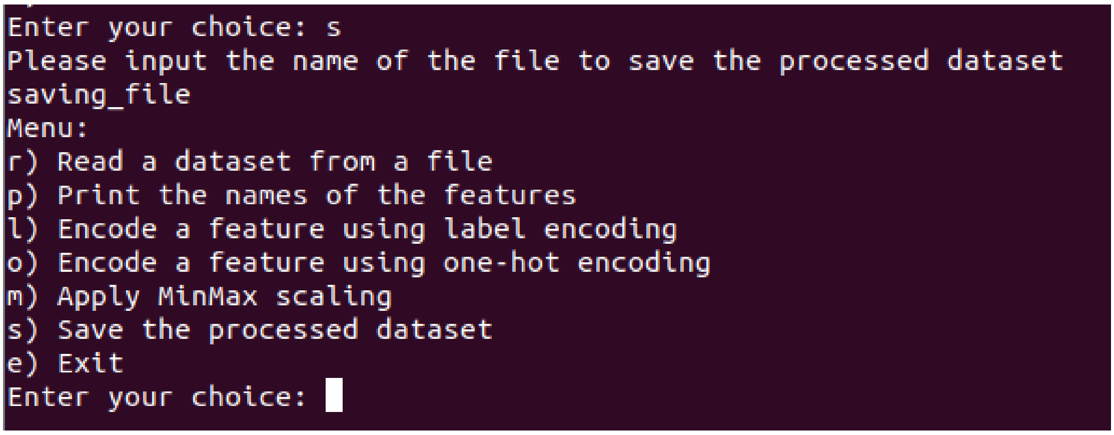
MinMax scaling verify if the feature is encoded or it's a numerical, if it's not the program will print a message that must be encoded first.



```
Enter your choice: m
Please input the name of feature to be scaled:
height
===============================
0.35
0.00
0.17
1.00
0.21
[0.35,0.00,0.17,1.00,0.21]
===============================
Minimum value: 160
Maximum value: 188
Menu:
r) Read a dataset from a file
p) Print the names of the features
l) Encode a feature using label encoding
o) Encode a feature using one-hot encoding
m) Apply MinMax scaling
s) Save the processed dataset
e) Exit
Enter your choice:
```

Here the output of MinMax scaling if we choose for example height feature, it will print the minimum,maximum values and apply the MinMax scaling to the feature vector.

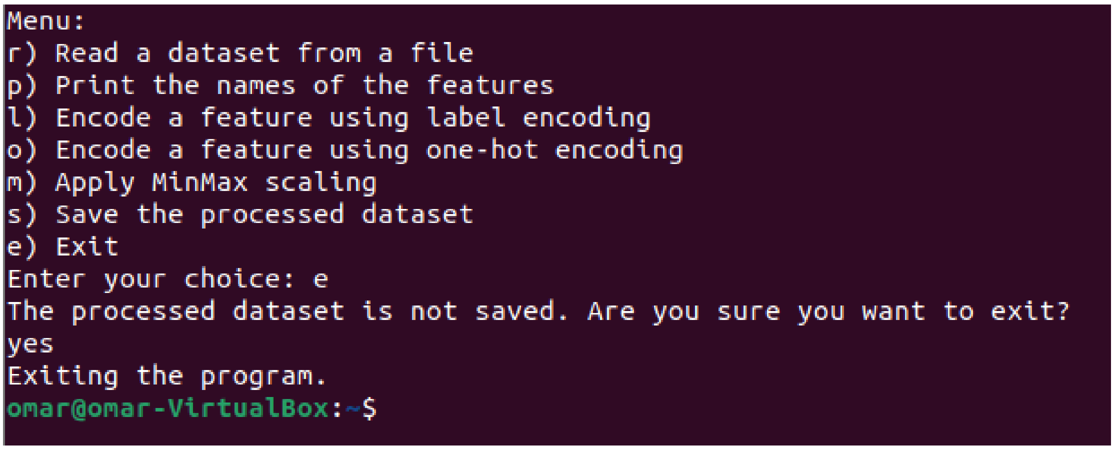
Save option



```
Enter your choice: s
Please input the name of the file to save the processed dataset
saving_file
Menu:
r) Read a dataset from a file
p) Print the names of the features
l) Encode a feature using label encoding
o) Encode a feature using one-hot encoding
m) Apply MinMax scaling
s) Save the processed dataset
e) Exit
Enter your choice: █
```

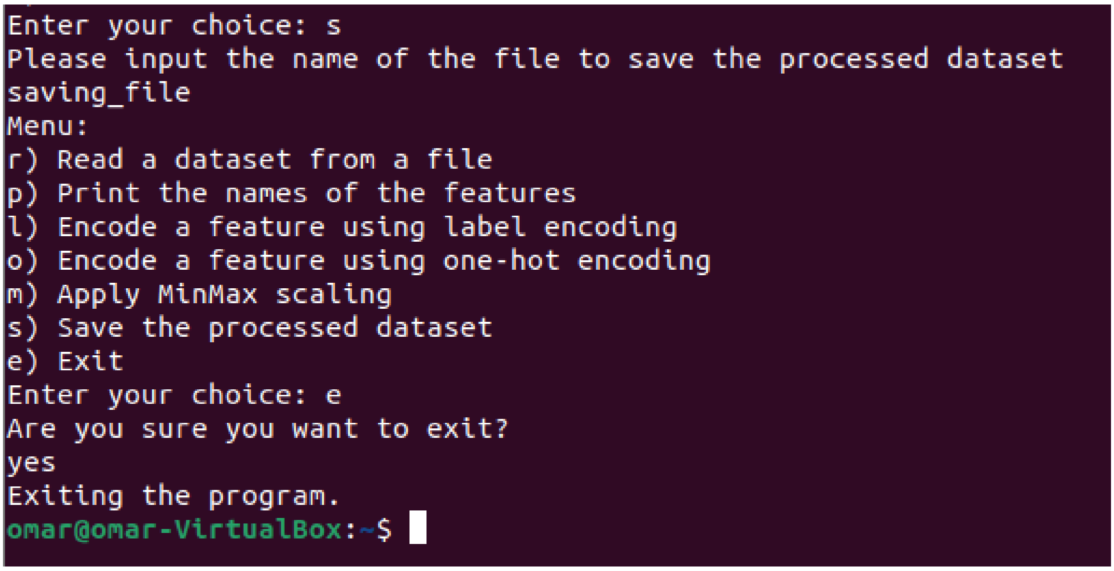
If the user enters (s) in the menu, the script should save the processed dataset to a file.

Exit option



```
Menu:
r) Read a dataset from a file
p) Print the names of the features
l) Encode a feature using label encoding
o) Encode a feature using one-hot encoding
m) Apply MinMax scaling
s) Save the processed dataset
e) Exit
Enter your choice: e
The processed dataset is not saved. Are you sure you want to exit?
yes
Exiting the program.
omar@omar-VirtualBox:~$
```

The program should check if the processed dataset is saved, if not, the program should print message on the screen that the processed dataset not saved, are you sure you want to exit?



```
Enter your choice: s
Please input the name of the file to save the processed dataset
saving_file
Menu:
r) Read a dataset from a file
p) Print the names of the features
l) Encode a feature using label encoding
o) Encode a feature using one-hot encoding
m) Apply MinMax scaling
s) Save the processed dataset
e) Exit
Enter your choice: e
Are you sure you want to exit?
yes
Exiting the program.
omar@omar-VirtualBox:~$
```

If the dataset is saved, the program should print on the screen “Are you sure you want to exist”. If the user inputs “yes”, the program ends, else will back to menu.

CODE:

```
#!/bin/bash

# Flag to track whether a dataset has been read from a file

dataset_read=false

# Flag to track whether a dataset has been saved

dataset_saved=false

# Main menu loop

while true; do

  # Print the menu

  echo "Menu:"

  echo "r) Read a dataset from a file"

  echo "p) Print the names of the features"

  echo "l) Encode a feature using label encoding"

  echo "o) Encode a feature using one-hot encoding"

  echo "m) Apply MinMax scaling"

  echo "s) Save the processed dataset"
  echo "e) Exit"
  # Read the user's choice
  echo -n "Enter your choice: "
  read choice
  # Perform the selected action
  case $choice in
    r)
      # Action for reading a dataset from a file
      echo "Please input the name of the dataset file: "
      read file_name
      if [ ! -f "$file_name" ]; then
          echo "File does not exist."
      else
                if [ ! -f "temp.txt" ]; then
                     touch "temp.txt"
               fi
               #copy the contents of a file to a new file
      cat "$file_name" > "new_file.txt"

          line_count=$(wc -l < $file_name)

          header_line=$(head -n 1 $file_name)

          header_line=$(echo "$header_line" | sed 's/[[:space:]]*$//')

          first_line=$(head -n 1 $file_name | tr -s ";" " " | cut -d" " -f1-)

          line1_count=$(echo $first_line |wc -w)

           second_line=$(tail -n +2 $file_name | head -n 1 |tr -s ";" " " | cut -d" " -f1- | wc -w)

          # Code to check the format of the data in the dataset file goes here

          if [ "$second_line" !=  "$line1_count" ]; then

          echo "The format of the data in the dataset file is wrong."

          break

          else

          echo "The file has been read"

          fi

          declare -a minmax_array

          # If the format is correct, set the dataset_read flag to true
```

```
                    dataset_read=true
      fi
;;


    p)
      # Action for printing the names of the features


      if [ "$dataset_read" = false ]; then
        echo "You must first read a dataset from a file."
      else
      echo "************************************************************"


          echo "$header_line" | tr ";" '  '


      echo "************************************************************"


          fi
      ;;


    l)


      # Action for label encoding a feature


      if [ "$dataset_read" = false ]; then


        echo "You must first read a dataset from a file."


      else


      echo "Please input the name of the categorical feature for label encoding: 
"


        read feature_name
```

```
#clear the file

>"temp.txt"

        # Set up a flag to track whether featurename was found

        found=false

        #to count the number of column

        counter=0

        features=$(echo $first_line | tr ";" '\n')

        for feature in $features

        do

        counter=$((counter+1))

        if [ "$feature_name" = "$feature" ]; then

               # If the value is not in the array, add it and assign it a new code

         if [[ ! " ${minmax_array[*]} " =~ " $feature_name" ]]; then

        #add the feature to min-max array

        minmax_array+=($feature_name)

    fi

        found=true
    break
fi
done
    header_line=$(head -n 1 $file_name)
    echo "$header_line" > "temp.txt"
    if $found
    then
    # Create a dictionary
    declare -A value_code
    declare -A val
    code=0
    line_count=0
    while read line; do
     line_count=$((line_count+1))
    # Skip the first line that contain the sataset
    if [ "$line_count" -eq 1 ]; then
    continue
```

```
fi
values=$(echo $line |cut -d";" -f$counter)
 for value in $values; do
 if [ -z "${value_code[$value]}" ]; then
# If the value is not in the dictionary, add it and assign it a new code
value_code[$value]=$code
code=$((code + 1))
fi
if [ -z "${val[$value]}" ]; then
val[$value]=$value
fi
#set a new values
modified_line=$(sed "s/${val[$value]}/${value_code[$value]}/g" <<< "$line")
echo "$modified_line" >> "temp.txt"
done
done <"new_file.txt"
cat "temp.txt" > "new_file.txt"
```

```
        # to access all the elements

        for key in "${!value_code[@]}"; do

        echo "Value: $key, Code: ${value_code[$key]}"

        done

        else

        echo "The name of categorical feature is wrong"

        fi

        label_encoded=true

        fi
    ;;

   o)

    # Action for one-hot encoding a feature

    if [ "$dataset_read" = false ]; then

      echo "You must first read a dataset from a file."
    else
    echo "Please input the name of the categorical feature for one-hot
encoding: "

      read feature_name

      # Set up a flag to track whether featurename was found
      found=false

#clear file

>"temp.txt"

      #to count the number of column

      counter=0
      declare -a header_array

      features=$(echo $first_line | tr ";" '\n')

      for feature in $features

      do

      counter=$((counter+1))

      if [ "$feature_name" = "$feature" ]; then

      header_array+=($feature)

      # If the value is not in the dictionary, add it and assign it a new code

          if [[ ! " ${minmax_array[*]} " =~ " $feature_name" ]]; then

        minmax_array+=($feature_name)

    fi

      found=true
```

```
        break

fi

done

        header_line=$(head -n 1 $file_name)

header_line=$(echo "$header_line" | sed 's/[[:space:]]*$//' | sed "s/$feature_name;//")

        if $found

        then

        # Create a dictionary

        declare -a values_array

           code=0

        line_count=0

        while read line; do

         line_count=$((line_count+1))

        # Skip the first line that contain the sataset

        if [ "$line_count" -eq 1 ]; then

        continue
```

```
fi
values=$(echo $line |cut -d";" -f$counter)
 for value in $values; do
  if [[ ! " ${values_array[*]} " =~ " $value " ]]; then
   values_array+=($value)
fi
   done
   str=$(IFS=';'; echo "${values_array[*]}")
   done < "new_file.txt"
   header_line="$header_line$str;"
   echo "$header_line" > "temp.txt"
values=$(echo $line |cut -d";" -f$counter)
   num=0
   while read line; do
   num=$((num+1))
values=$(echo $line |cut -d";" -f$counter)
   # initialize array encoded data
   array=()
```

```
          for val in "${values_array[@]}"; do

          if [ "$val" == "$values" ]; then

          array+=("1;")
  else
       array+=("0;")

  fi

done

          oneHot_data=""
          for i in "${array[@]}"; do
          oneHot_data+="$i"

          done
          if [ "$num" -ge 2 ]; then
            if [ "$num" -ge "$line_count" ]; then

            break

            fi
            line=$(echo "$line" | sed 's/[[:space:]]*$//' | sed "s/$values;//")

          modified_line="$line$oneHot_data"

       echo "$modified_line" >> "temp.txt"

   fi
       done <"new_file.txt"

       cat "temp.txt" > "new_file.txt"
       else
       echo "The name of categorical feature is wrong"
```

```
        fi
          oneHot_encoded=true


        fi


      ;;


    m)


      # Action for applying MinMax scaling
          if [ "$dataset_read" = false ]; then
        echo "You must first read a dataset from a file."
      else
          echo "Please input the name of feature to be scaled: "
        read feature_name
        find=false
        checked=false
        count=0


>"temp.txt"
        l_count=0
        featur=$(echo $first_line | tr ";" '\n')
          for feature in $featur
          do


          count=$((count+1))
 if [ "$feature_name" = "$feature" ]; then
        find=true
        break
     fi
          done


if $find
          then


         #check if the entered feature are encoded
          for key in "${minmax_array[@]}"; do
          if [ "$feature_name" = "$key"  ]; then
          checked=true
        fi
        done

        values=$(tail -n +2 "new_file.txt" |cut -d";" -f$count)

          for value in "${values[@]}"; do

          #check if the feature is numeric

          if [[ -z "`echo "$value" | sed 's/./\0\n/g' | grep -v [0-9] | tr -d '\n'`" ]]; then

          checked=true

          fi
          done

    if $checked; then

        # Initialize the minimum and maximum values to the first element of the
array

          min=${values[0]}
          max=${values[0]}



          # finds the minimum and maximum values in a list of values

          min=`echo $values | tr ' ' '\n' | sort -n | head -1`

          max=`echo $values | tr ' ' '\n' | sort -n | tail -1`

        arr=()
```

```
  dm=$((max-min))

  echo "=================================="

    values=(`echo "$values"`)

    for value in "${values[@]}"; do


            vi=$(echo "scale=2;$value-$min" | bc -l)

            res=$(echo "scale=2;$vi/$dm" | bc -l | awk '{printf "%.2f\n", $0}')

            echo "$res"

            arr+=($res)

    done

    #print the array that contain scaled feature

    echo $(echo "[${arr[@]}]" | tr ' ' ,)

  echo "=================================="

    # Print the minimum and maximum values

    echo "Minimum value: $min"

    echo "Maximum value: $max"

else
```

```
         echo "This feature is categorical feature and must be encoded first "
      fi
        else
        echo "Feature not found"
        fi
          fi
          ;;
    s)
      # Action for saving the processed dataset
      if [ "$dataset_read" = false ]; then
          echo "The processed dataset is not saved. Are you sure you want to
exist"


      else
        echo "Please input the name of the file to save the processed dataset"
          read filename
          if [ ! -f "$filename" ]; then

                  touch $filename

           fi

          #copy the data to file for saving

          cat "new_file.txt" >> $filename

          #change the flag of save

          dataset_saved=true

      fi

      ;;

    e)

      # Exit the program

      if [ "$dataset_saved" = false ]; then

      echo "The processed dataset is not saved. Are you sure you want to
exit?"

      read confrim1

      if [ "$confrim1" = "yes" ];then
      echo "Exiting the program."
      exit
      fi
```

```
      else

      echo "Are you sure you want to exit?"

      read confrim2

      if [ "$confrim2" = "yes" ];then

      echo "Exiting the program."

      exit

      fi


      fi

      ;;

     *)

      # Invalid choice

      echo "Invalid choice. Please try again."

          ;;

esac

done
```

Dataset:

id;age;gender;height;weight;active;smoke;governorate;

1;30;male;170;88;no;yes;ramallah;

2;25;female;160;65;no;no;ramallah;

3;28;male;165;72;yes;yes;nablus;

4;44;male;188;90;no;no;jerusalem;

5;60;female;166;70;no;no;jerusalem;

age;sex;bmi:children;smoker;region;charges;

18;male;33;1;no;southeast;1725;

28;male;33;3:no;southeast;4449;

32;male;28;0;no;northwest;3866;

46;female;33;1;yes;southeast;8240;

45;male;38;2;no;northwest;6866;

63;female;52;4;no;southeast;9650;