جامعة بيرزيت

BIRZEIT UNIVERSITY

Department of Electrical and Computer Engineering

Linux lab

Project (1)

Dataset preprocessing

Student’s names & Id’s :

Rama Abdlrahman 1191344

Dana Hammad 1191568

Instructor: D. Mohammad Jubran

Date: 29/12/2022

- ## Main Idea

Since the dataset is **"clean".** We only need to take care of the logic of the operations requested such as **encoding and scaling** without having to worry about too many edge cases that a non-clean dataset would create. Both types of encoding follow the same logic except **one-hot encoding** has some extra steps. This means that implementing them together in the same function would shorten the code.

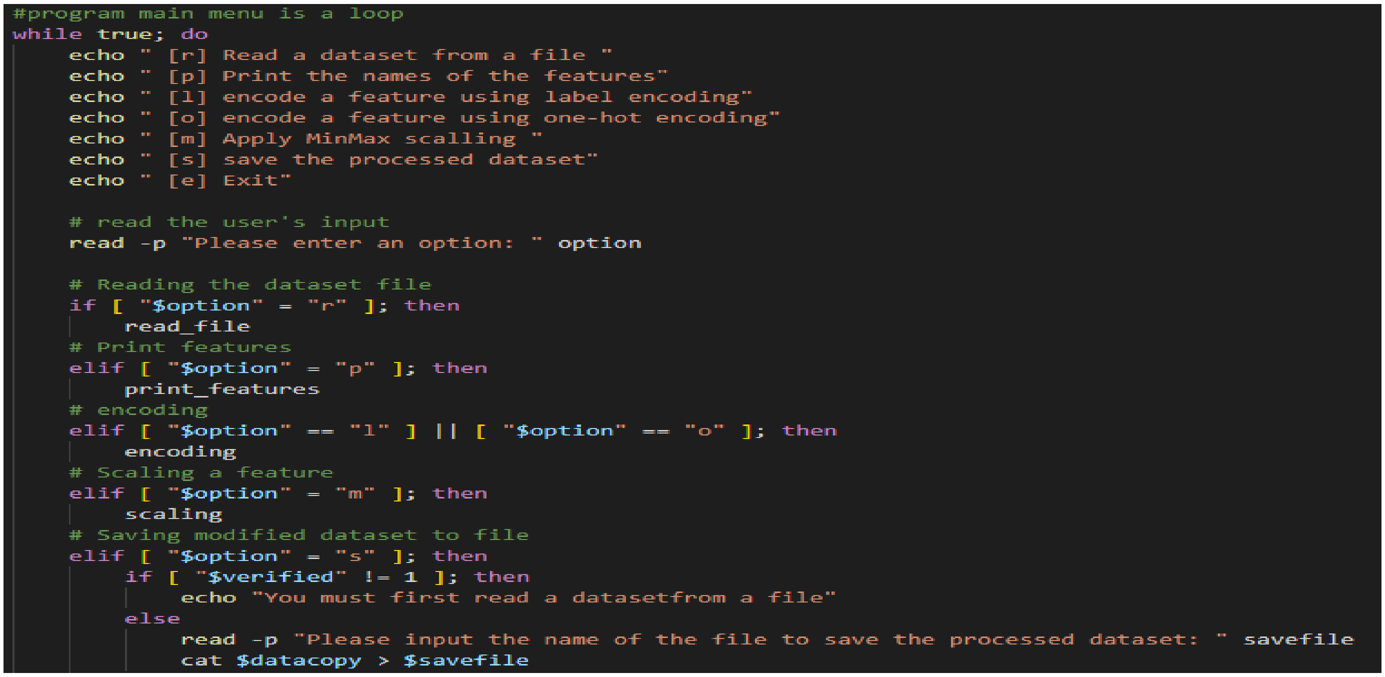
We can use **cut and paste commands** to extract columns of data from the dataset easily, do the encoding and scaling required and then put them back into the original dataset.

- **Explanation and test cases:**

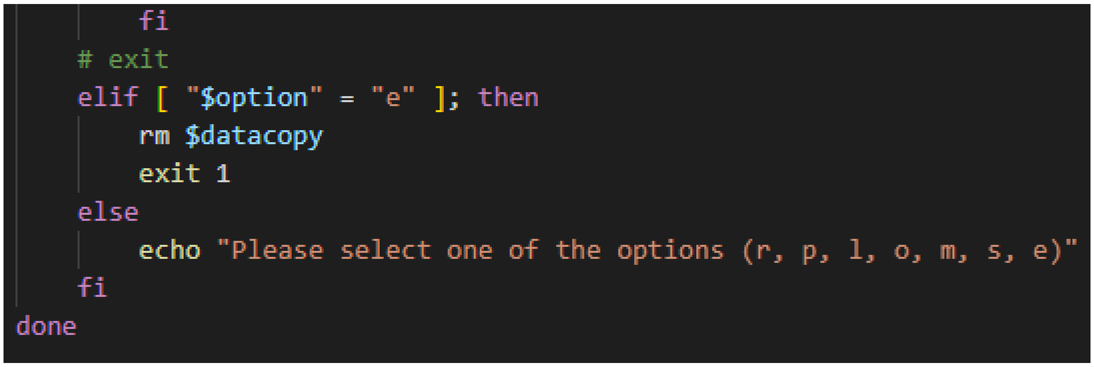
1. **The main menu:**

The main menu is a while loop that keeps on printing the menu lines and asking for user input. We check what the input is and run a certain function for each case. For example, if the option is “r” then we run the “read_file” function.

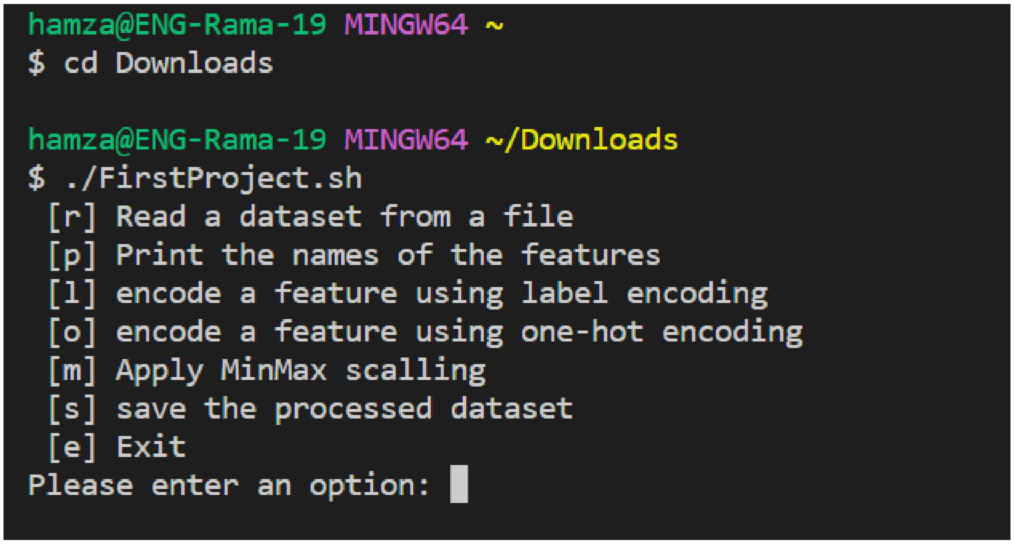
The encoding function holds the logic for both label encoding and one-hot encoding so we run it for both the “l” and “o” options (we check which one it is later inside the function).



```
#program main menu is a loop
while true; do
    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 scalling "
    echo " [s] save the processed dataset"
    echo " [e] Exit"

    # read the user's input
    read -p "Please enter an option: " option

    # Reading the dataset file
    if [ "$option" = "r" ]; then
        read_file
    # Print features
    elif [ "$option" = "p" ]; then
        print_features
    # encoding
    elif [ "$option" == "l" ] || [ "$option" == "o" ]; then
        encoding
    # Scaling a feature
    elif [ "$option" = "m" ]; then
        scaling
    # Saving modified dataset to file
    elif [ "$option" = "s" ]; then
        if [ "$verified" != 1 ]; then
            echo "You must first read a datasetfrom a file"
        else
            read -p "Please input the name of the file to save the processed dataset: " savefile
            cat $datacopy > $savefile
```

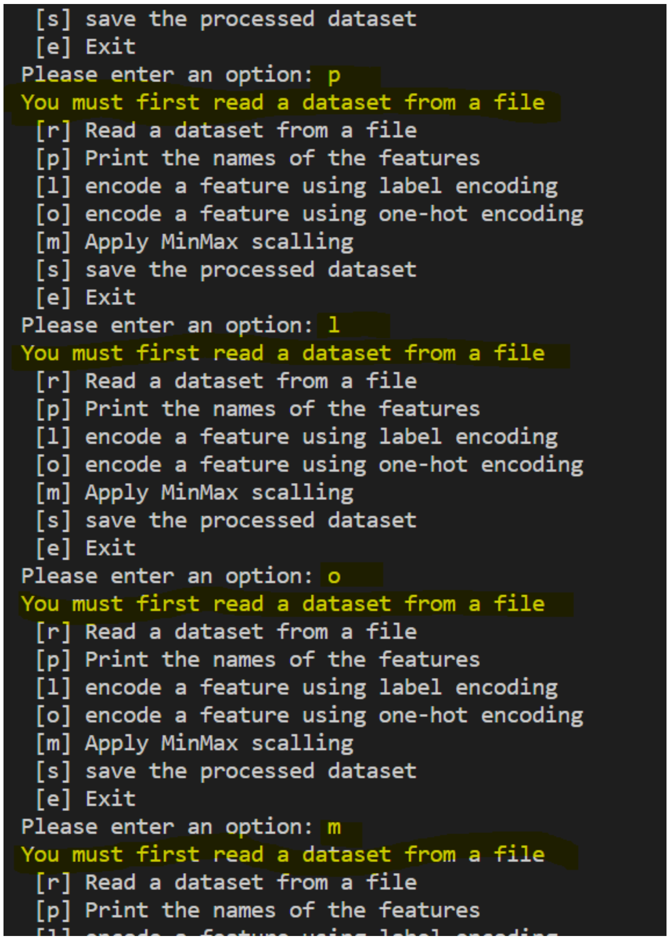


```
        fi
    # exit
    elif [ "$option" = "e" ]; then
        rm $datacopy
        exit 1
    else
        echo "Please select one of the options (r, p, l, o, m, s, e)"
    fi
done
```

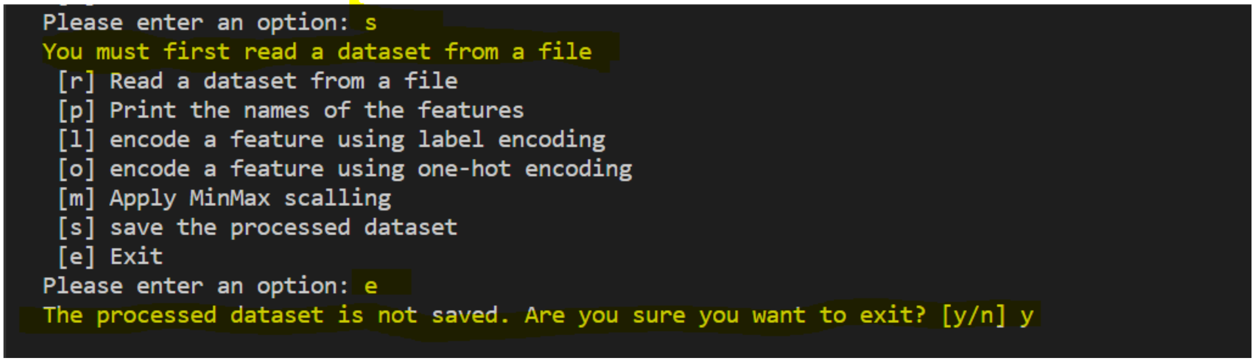


```
hamza@ENG-Rama-19 MINGW64 ~
$ cd Downloads

hamza@ENG-Rama-19 MINGW64 ~/Downloads
$ ./FirstProject.sh
 [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 scalling
 [s] save the processed dataset
 [e] Exit
Please enter an option: 
```

At first , if you entered anything except ‘r’ or ‘e’ , program will print you a message that you should read file first and the menu will appear again:

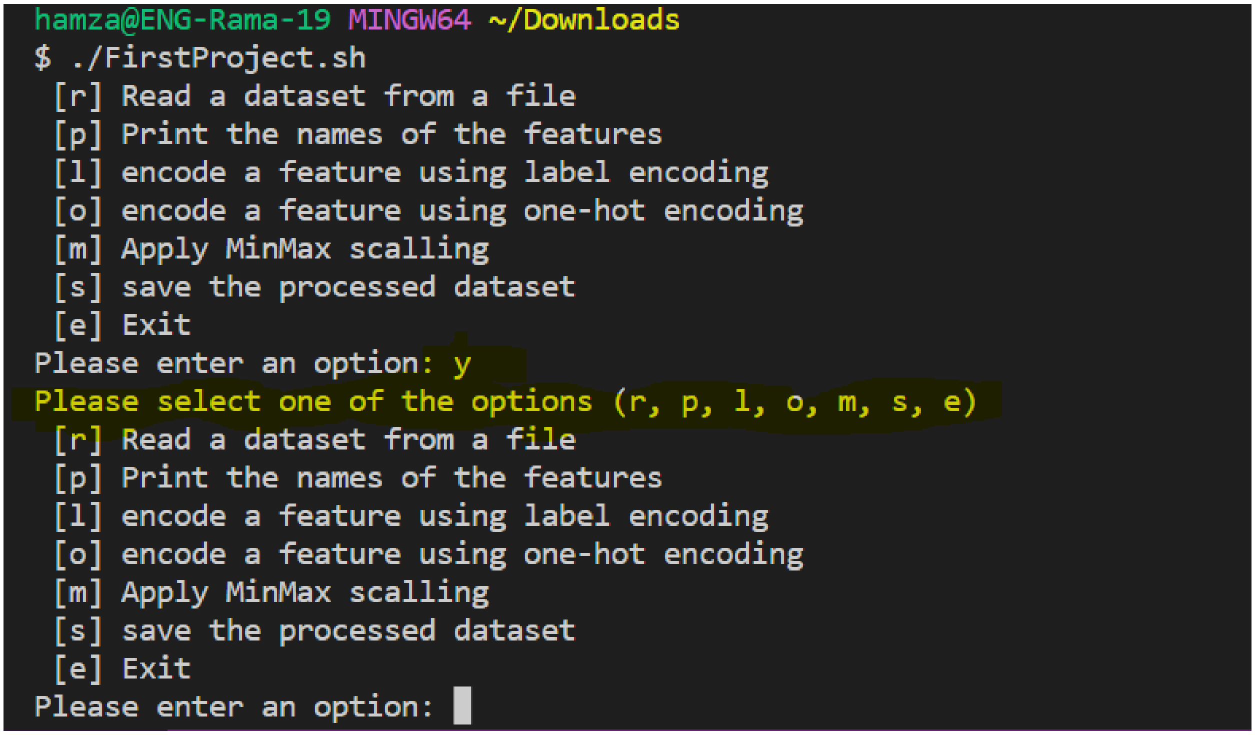


```
 [s] save the processed dataset
 [e] Exit
Please enter an option: p
You must first read a dataset from a file
 [r] Read a dataset from a file
 [p] Print the names of the features
 [l] encode a feature using label encoding
 [o] encode a feature using one-hot encoding
 [m] Apply MinMax scalling
 [s] save the processed dataset
 [e] Exit
Please enter an option: l
You must first read a dataset from a file
 [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 scalling
 [s] save the processed dataset
 [e] Exit
Please enter an option: o
You must first read a dataset from a file
 [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 scalling
 [s] save the processed dataset
 [e] Exit
Please enter an option: m
You must first read a dataset from a file
 [r] Read a dataset from a file
 [p] Print the names of the features
 [l] encode a feature using label encoding
```



```
Please enter an option: s
You must first read a dataset from a file
 [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 scalling
 [s] save the processed dataset
 [e] Exit
Please enter an option: e
The processed dataset is not saved. Are you sure you want to exit? [y/n] y
```

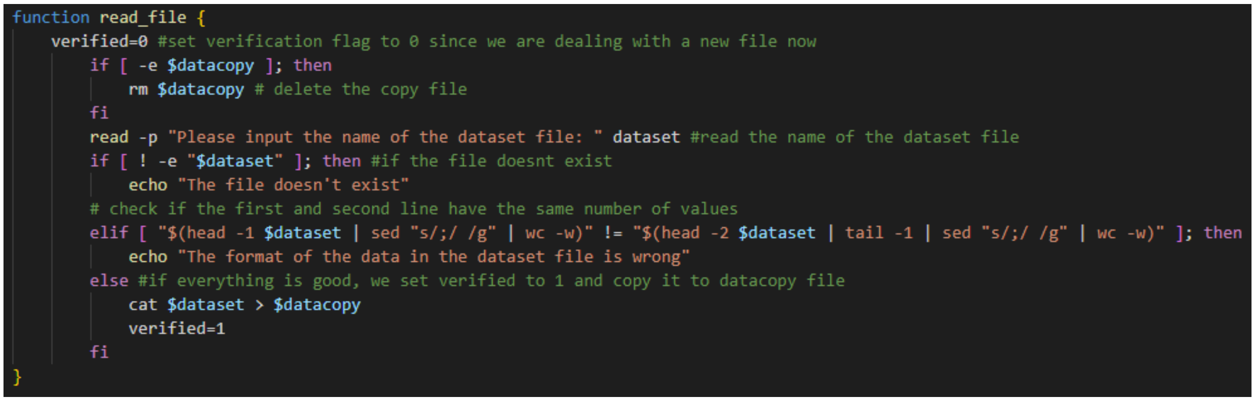
If you entered wrong choice, a statement asks you to select one of the options in the menu will be printed , and the menu will appear again:



```
hamza@ENG-Rama-19 MINGW64 ~/Downloads
$ ./FirstProject.sh
 [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 scalling
 [s] save the processed dataset
 [e] Exit
Please enter an option: y
Please select one of the options (r, p, l, o, m, s, e)
 [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 scalling
 [s] save the processed dataset
 [e] Exit
Please enter an option: █
```

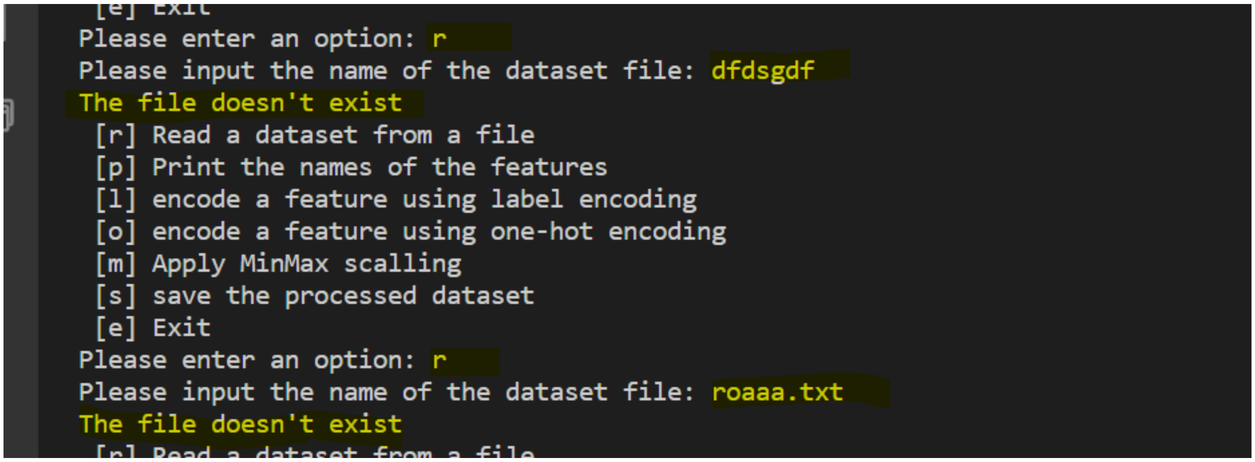
2. Reading the file:

To read the file, we first ask the user to input the name of the file. Then, we check if the file exists. After that, we check if the file is file is valid by comparing the second line’s number of columns first line’s. If they’re the same, we set the verified flag to 1 to indicate that the file has been verified. Then, we copy the file to a new file using cat which is where we will do all the other operations necessary (the original input file remains the same).



```
function read_file {
    verified=0 #set verification flag to 0 since we are dealing with a new file now
        if [ -e $datacopy ]; then
            rm $datacopy # delete the copy file
        fi
        read -p "Please input the name of the dataset file: " dataset #read the name of the dataset file
        if [ ! -e "$dataset" ]; then #if the file doesnt exist
            echo "The file doesn't exist"
        # check if the first and second line have the same number of values
        elif [ "$(head -1 $dataset | sed "s/;/ /g" | wc -w)" != "$(head -2 $dataset | tail -1 | sed "s/;/ /g" | wc -w)" ]; then
            echo "The format of the data in the dataset file is wrong"
        else #if everything is good, we set verified to 1 and copy it to datacopy file
            cat $dataset > $datacopy
            verified=1
        fi
}
```

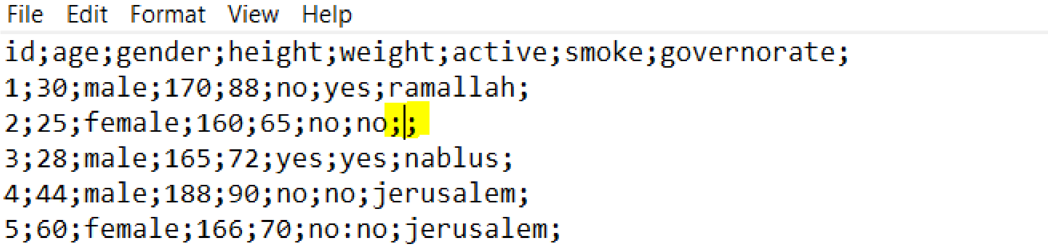
<u>verifying if the file exists:</u>



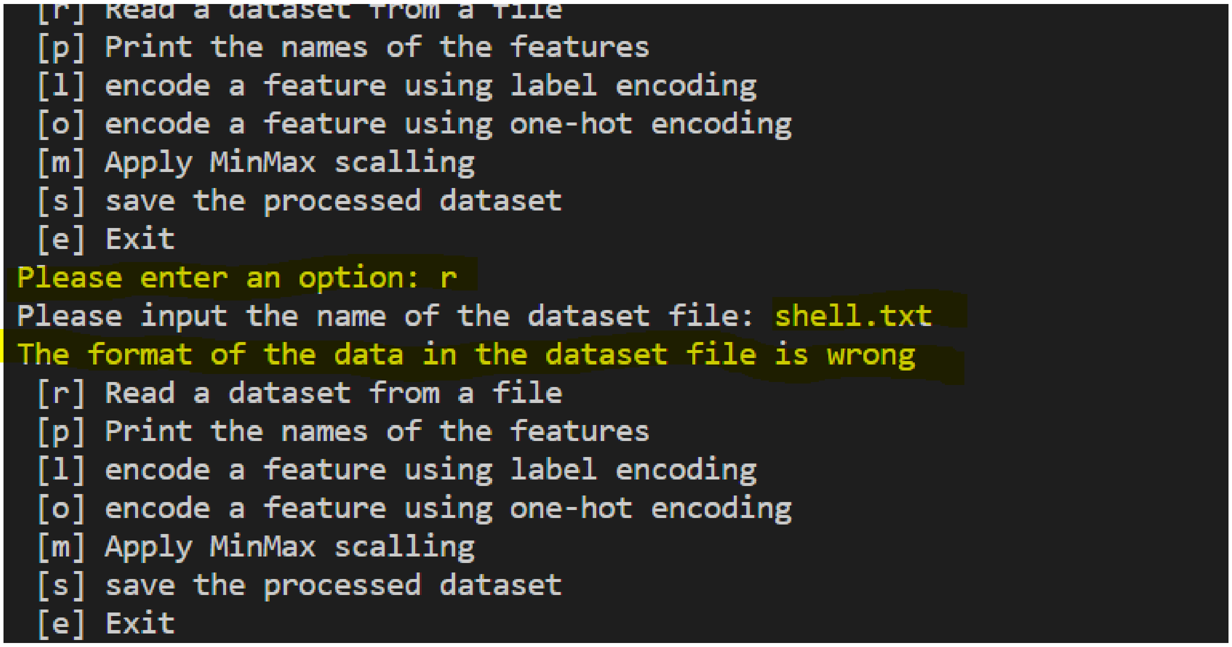
```
  [e] Exit
Please enter an option: r
Please input the name of the dataset file: dfdsgdf
The file doesn't exist
  [r] Read a dataset from a file
  [p] Print the names of the features
  [l] encode a feature using label encoding
  [o] encode a feature using one-hot encoding
  [m] Apply MinMax scalling
  [s] save the processed dataset
  [e] Exit
Please enter an option: r
Please input the name of the dataset file: roaaa.txt
The file doesn't exist
  [r] Read a dataset from a file
```

<u>checking the format of the data in the dataset file. In case of any format problems, the program should print on the screen “The format of the data in the dataset file is wrong” and then return to the main menu:</u>

<u>example of wrong format:</u>



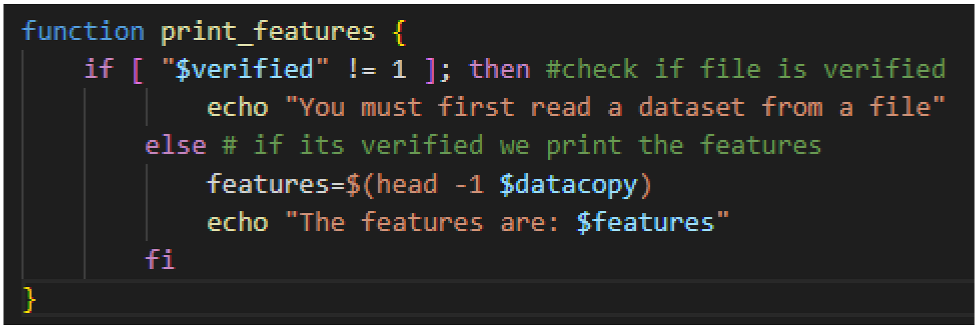
```
id;age;gender;height;weight;active;smoke;governorate;
1;30;male;170;88;no;yes;ramallah;
2;25;female;160;65;no;no;;
3;28;male;165;72;yes;yes;nablus;
4;44;male;188;90;no;no;jerusalem;
5;60;female;166;70;no:no;jerusalem;
```



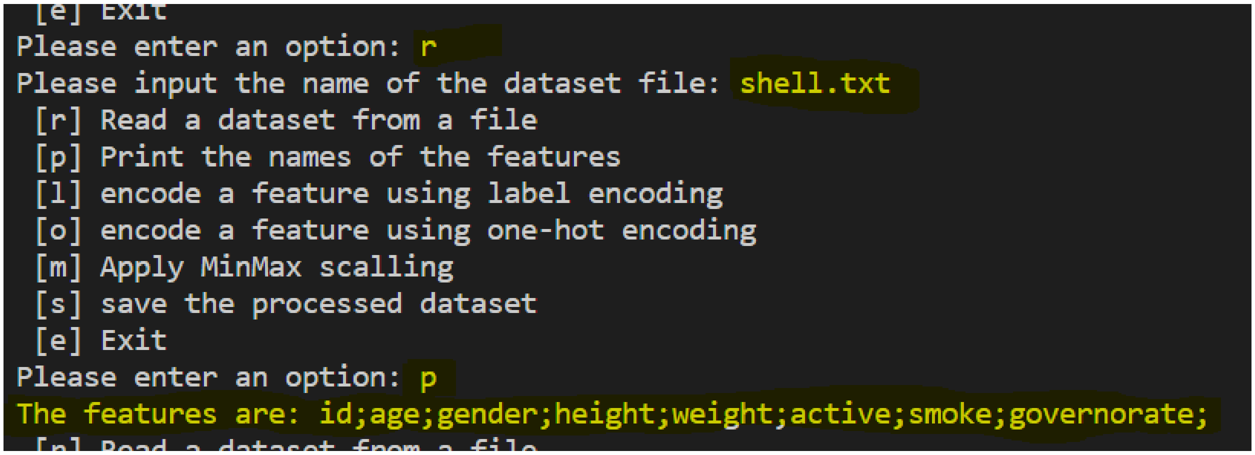
```
 [r] Read a dataset from a file
 [p] Print the names of the features
 [l] encode a feature using label encoding
 [o] encode a feature using one-hot encoding
 [m] Apply MinMax scalling
 [s] save the processed dataset
 [e] Exit
Please enter an option: r
Please input the name of the dataset file: shell.txt
The format of the data in the dataset file is wrong
 [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 scalling
 [s] save the processed dataset
 [e] Exit
```

3. **Printing the features:**

Printing the features is simple. We check if the verified flag is set to 1 (if we already read a file or not) and we print the first line using the head command of it if so. Otherwise, we print an error message and return to main menu.

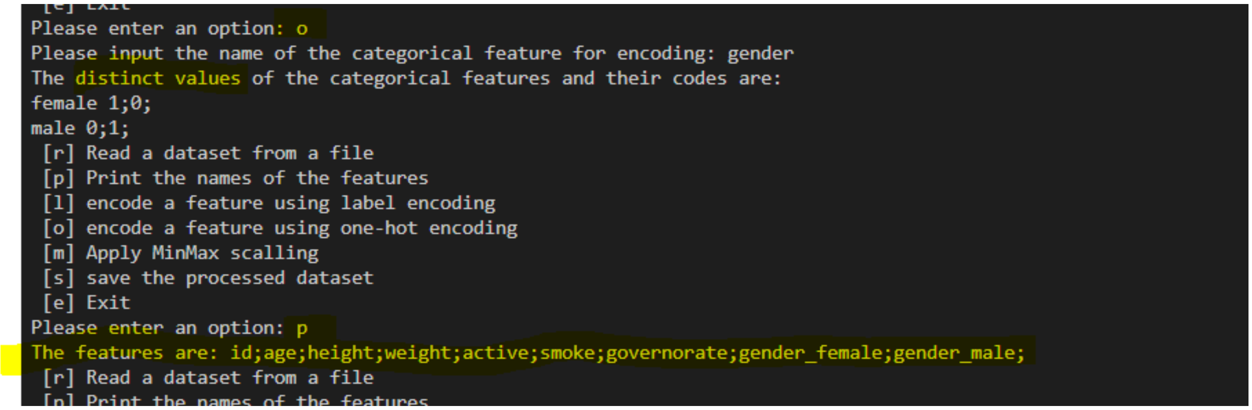


```
function print_features {
    if [ "$verified" != 1 ]; then #check if file is verified
            echo "You must first read a dataset from a file"
        else # if its verified we print the features
            features=$(head -1 $datacopy)
            echo "The features are: $features"
        fi
}
```



```
 [e] Exit
Please enter an option: r
Please input the name of the dataset file: shell.txt
 [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 scalling
 [s] save the processed dataset
 [e] Exit
Please enter an option: p
The features are: id;age;gender;height;weight;active;smoke;governorate;
 [r] Read a dataset from a file
```

Also printing features after one hot encoding on feature ‘gender’:

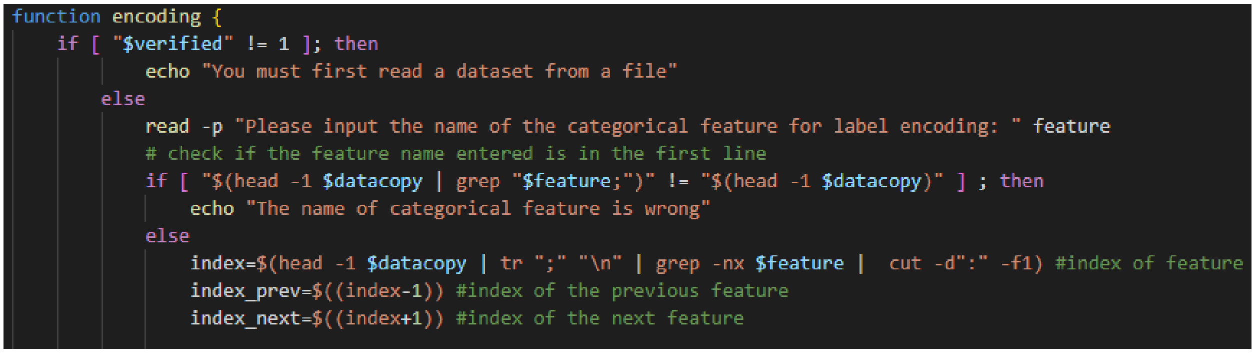


```
 [e] Exit
Please enter an option: o
Please input the name of the categorical feature for encoding: gender
The distinct values of the categorical features and their codes are:
female 1;0;
male 0;1;
 [r] Read a dataset from a file
 [p] Print the names of the features
 [l] encode a feature using label encoding
 [o] encode a feature using one-hot encoding
 [m] Apply MinMax scalling
 [s] save the processed dataset
 [e] Exit
Please enter an option: p
The features are: id;age;height;weight;active;smoke;governorate;gender_female;gender_male;
 [r] Read a dataset from a file
 [p] Print the names of the features
```

4. **Encoding:**

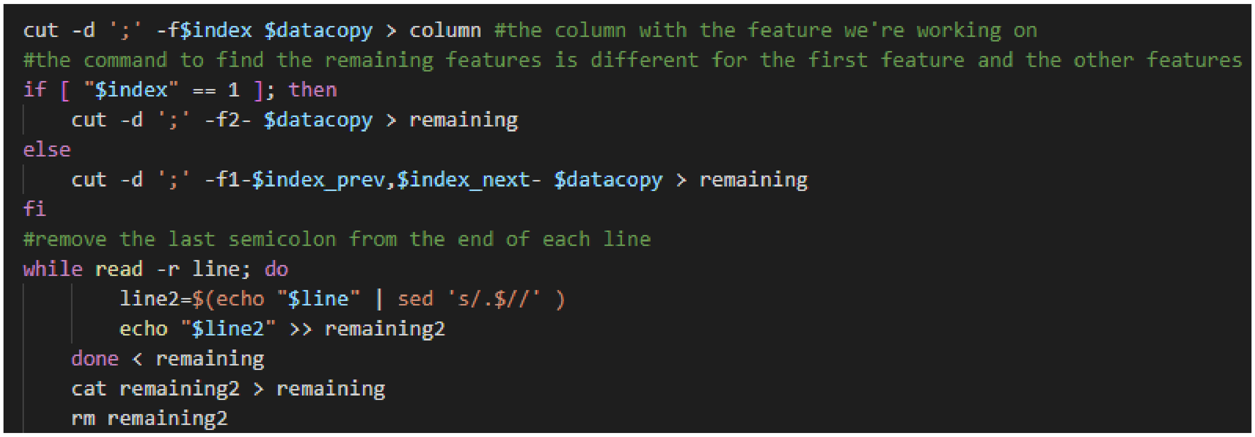
There are two kinds of encoding but like we mentioned earlier, they follow the same logic up to a certain point. That’s why they are included in the same function.

As before, we first check if the file is verified. Then, we ask the user for the name of the feature they want to encode and check if such feature exists. This is done by using grep and head to extract the feature name from the first line. If everything is good, we find the index of the feature by first using head to get the feature line, replacing the semicolons with newline characters, using grep -nx to print “linenumber:value” and finally cut using colon as a delimiter to extract the first value (line number) which is the index. Then, we calculate the index of the next feature and the one before it as seen below. This will be used in the cut command to extract the feature’s column.



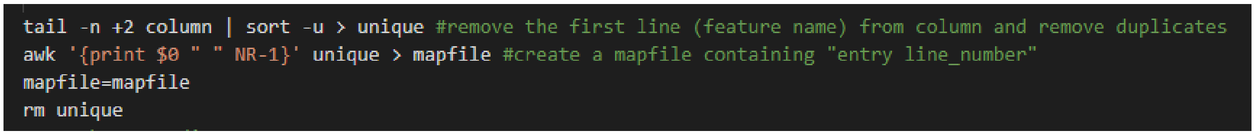
```
function encoding {
    if [ "$verified" != 1 ]; then
            echo "You must first read a dataset from a file"
        else
            read -p "Please input the name of the categorical feature for label encoding: " feature
            # check if the feature name entered is in the first line
            if [ "$(head -1 $datacopy | grep "$feature;")" != "$(head -1 $datacopy)" ] ; then
                echo "The name of categorical feature is wrong"
            else
                index=$(head -1 $datacopy | tr ";" "\n" | grep -nx $feature |  cut -d":" -f1) #index of feature
                index_prev=$((index-1)) #index of the previous feature
                index_next=$((index+1)) #index of the next feature
```

After that, we cut the feature column using semicolon as a delimiter and the index into a new file called column and we cut the remaining features into a file called remaining. We also remove the last semicolon from each line in the file since it caused us problems.



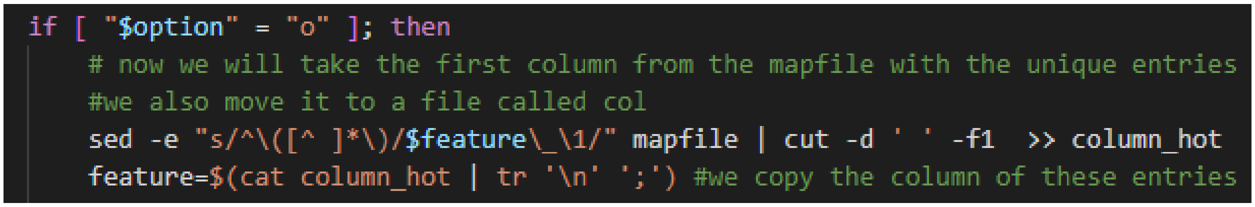
```
cut -d ';' -f$index $datacopy > column #the column with the feature we're working on
#the command to find the remaining features is different for the first feature and the other features
if [ "$index" == 1 ]; then
    cut -d ';' -f2- $datacopy > remaining
else
    cut -d ';' -f1-$index_prev,$index_next- $datacopy > remaining
fi
#remove the last semicolon from the end of each line
while read -r line; do
        line2=$(echo "$line" | sed 's/.$//' )
        echo "$line2" >> remaining2
    done < remaining
    cat remaining2 > remaining
    rm remaining2
```

To encode the file, we create a file called mapfile which holds the unique entries of the feature's column and their respective code. This is done by first removing the first line from the column file which holds the name of the feature, then sorting using sort -u which removes duplicate values and placing the result in a file called unique. Then, we use awk to print the values from the unique file with their line number into the mapfile. In the case of label encoding, the code is just the number of the line they're in.



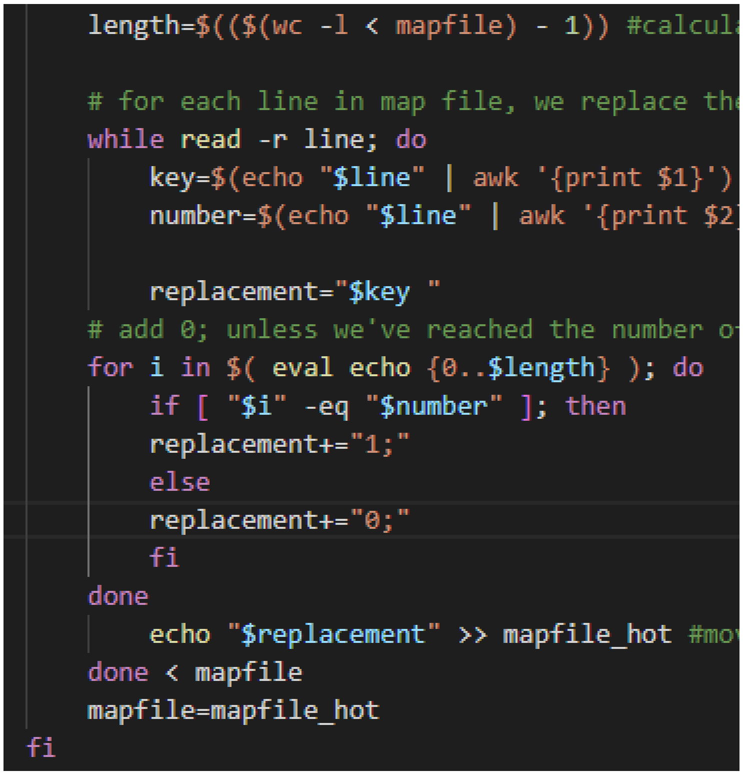
```
tail -n +2 column | sort -u > unique #remove the first line (feature name) from column and remove duplicates
awk '{print $0 " " NR-1}' unique > mapfile #create a mapfile containing "entry line_number"
mapfile=mapfile
rm unique
```

But in the case of one-hot encoding, we first need to modify the mapfile to hold the values of the unique entries of the feature's column but in the form "feature_value". This is done by using the sed command as seen below to the value which is cut from the mapfile's first column. After that, we remove the newline characters from the column and replace them with semicolons and store the value in the feature variable.



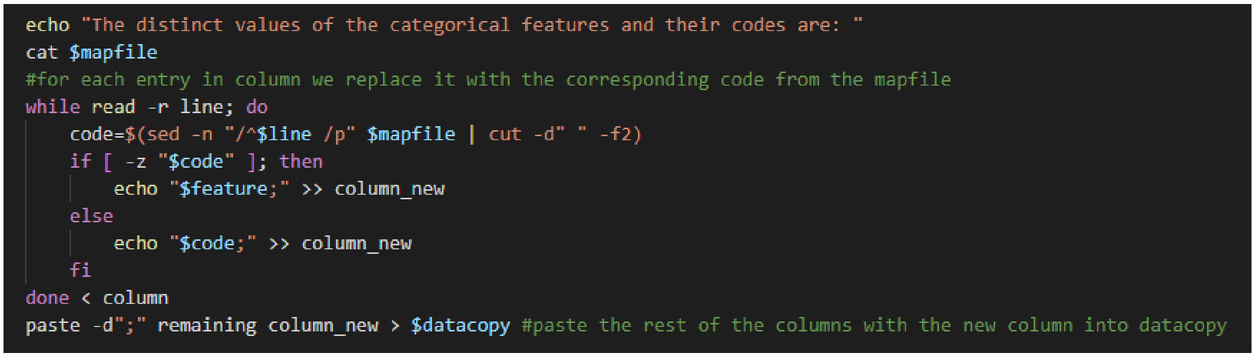
```
if [ "$option" = "o" ]; then
    # now we will take the first column from the mapfile with the unique entries
    #we also move it to a file called col
    sed -e "s/^\([^ ]*\)/$feature\_\1/" mapfile | cut -d ' ' -f1  >> column_hot
    feature=$(cat column_hot | tr '\n' ';') #we copy the column of these entries
```

Then, we calculate the length of the mapfile and we read it line by line, storing the value using awk from the first column (distinct value) into the key variable and the second column (code) into the number variable. We also create a variable called replacement which holds the key with a space. After that, we loop through a for loop that is as long as the length of the mapfile and we add 1; or 0; to replacement depending on the number of line we're on from the while loop and we store the result into a file called mapfile_hot which would now contain the one-hot code instead of the label code for the distinct values.



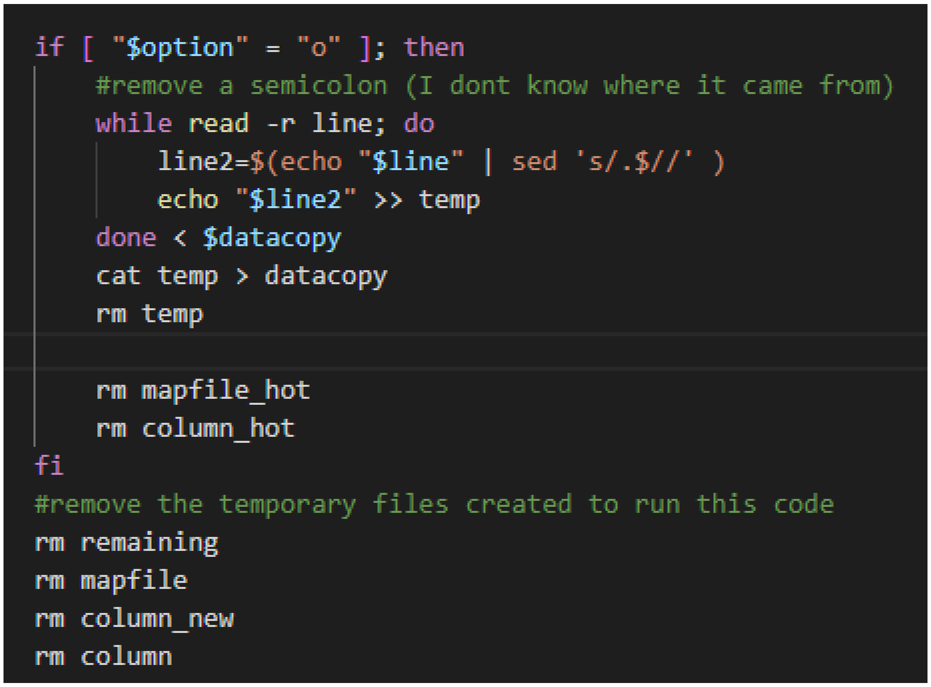
```
    length=$(($(wc -l < mapfile) - 1)) #calcul

    # for each line in map file, we replace the
    while read -r line; do
        key=$(echo "$line" | awk '{print $1}')
        number=$(echo "$line" | awk '{print $2

        replacement="$key "
    # add 0; unless we've reached the number o
    for i in $( eval echo {0..$length} ); do
        if [ "$i" -eq "$number" ]; then
        replacement+="1;"
        else
        replacement+="0;"
        fi
    done
        echo "$replacement" >> mapfile_hot #mo
    done < mapfile
    mapfile=mapfile_hot
fi
```

Following that, we print the data inside the mapfile using cat and we loop through the column file and create a new column replacing the value in the column with the corresponding value from the mapfile using sed to replace and cut to get the code from the mapfile. This works for both label encoding or one-hot encoding. We also check if the value of the code variable and if its null (we can’t find a code for a certain value). This means that we are on the feature name line and we just print it into the new column file. Finally, we paste the remaining file which contains the other columns with the new column into the datacopy file creating our new dataset with the encoded feature.



```
echo "The distinct values of the categorical features and their codes are: "
cat $mapfile
#for each entry in column we replace it with the corresponding code from the mapfile
while read -r line; do
    code=$(sed -n "/^$line /p" $mapfile | cut -d" " -f2)
    if [ -z "$code" ]; then
        echo "$feature;" >> column_new
    else
        echo "$code;" >> column_new
    fi
done < column
paste -d";" remaining column_new > $datacopy #paste the rest of the columns with the new column into datacopy
```

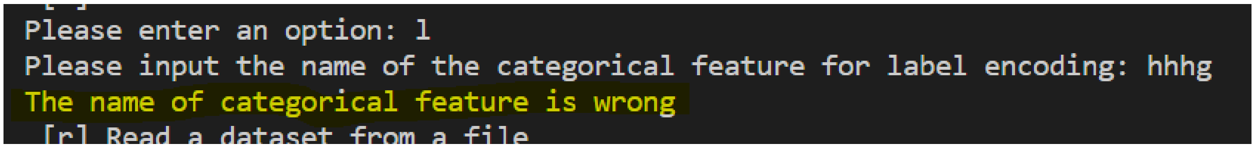
In addition, we remove an extra semicolon we get during one-hot encoding and we remove all of the files that we used for encoding.



```
if [ "$option" = "o" ]; then
    #remove a semicolon (I dont know where it came from)
    while read -r line; do
        line2=$(echo "$line" | sed 's/.$//' )
        echo "$line2" >> temp
    done < $datacopy
    cat temp > datacopy
    rm temp

    rm mapfile_hot
    rm column_hot
fi
#remove the temporary files created to run this code
rm remaining
rm mapfile
rm column_new
rm column
```

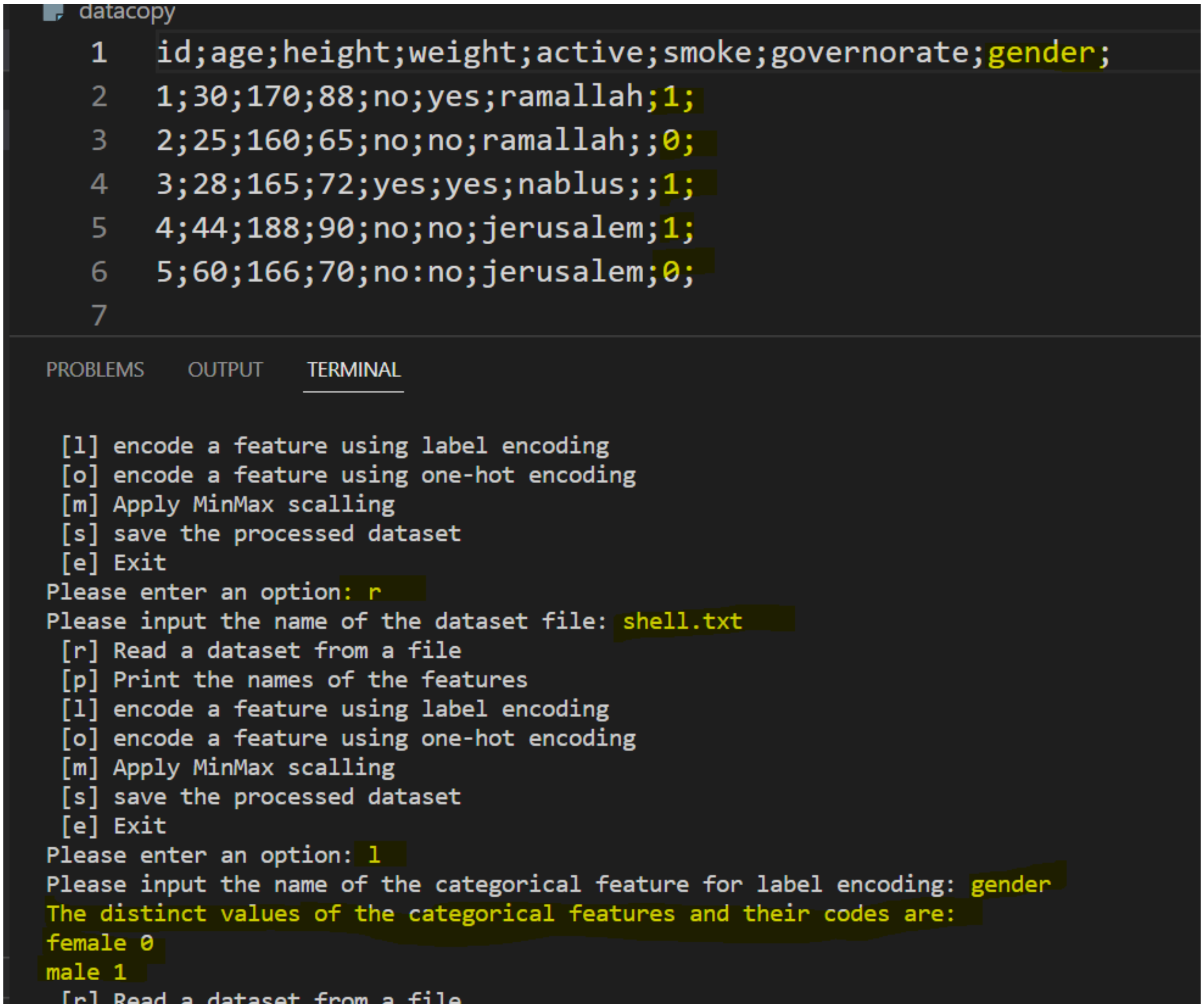
- **<u>Running for label encoding:</u>**

<u>program verifying that the entered name of the categorical feature exists in the dataset or not :</u>



```
Please enter an option: l
Please input the name of the categorical feature for label encoding: hhhg
The name of categorical feature is wrong
 [r] Read a dataset from a file
```

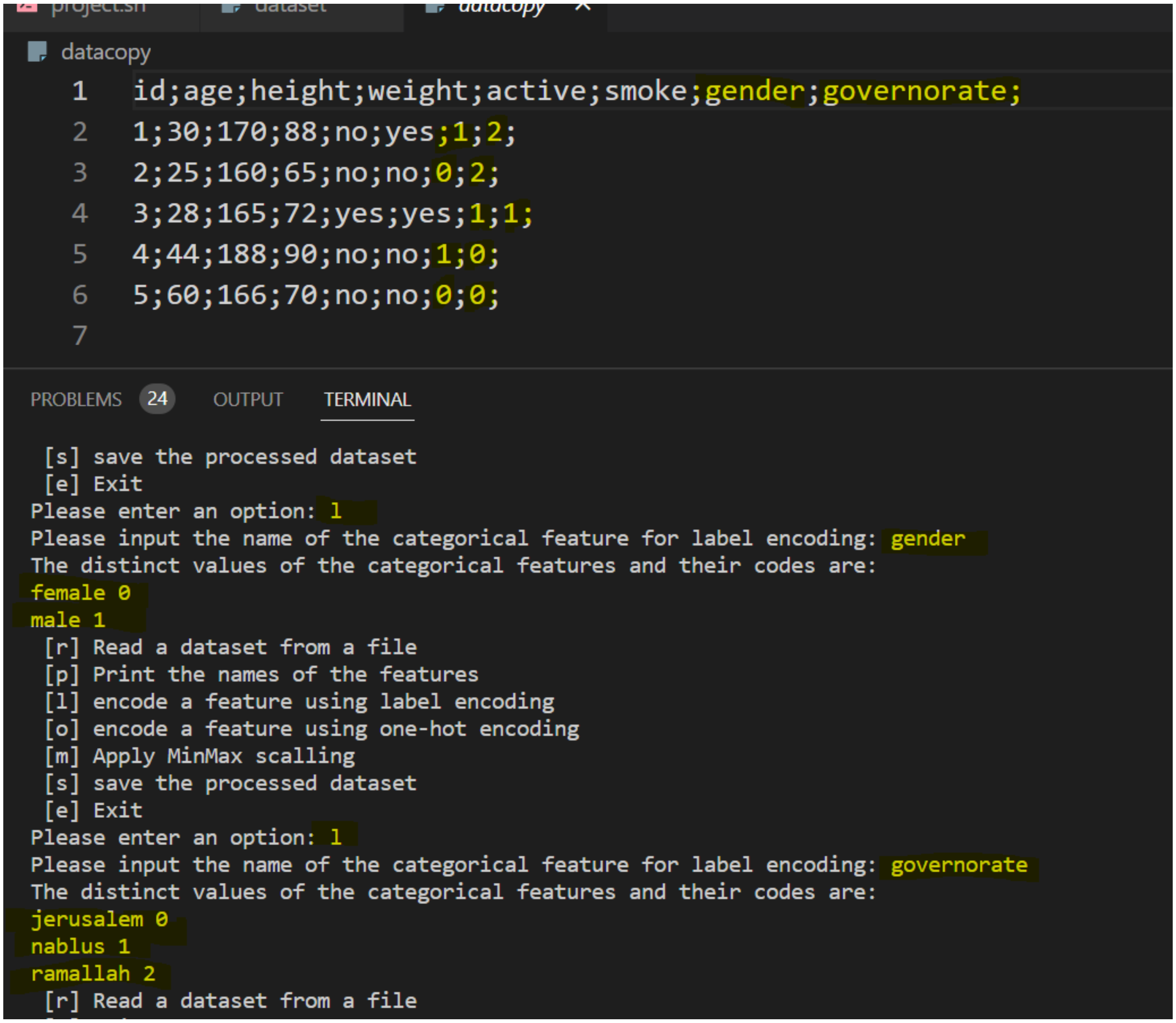
<u>Label encoding for ‘gender’ feature:</u>



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

```
 [l] encode a feature using label encoding
 [o] encode a feature using one-hot encoding
 [m] Apply MinMax scalling
 [s] save the processed dataset
 [e] Exit
Please enter an option: r
Please input the name of the dataset file: shell.txt
 [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 scalling
 [s] save the processed dataset
 [e] Exit
Please enter an option: l
Please input the name of the categorical feature for label encoding: gender
The distinct values of the categorical features and their codes are:
female 0
male 1
 [r] Read a dataset from a file
```

Label encoding for another feature:

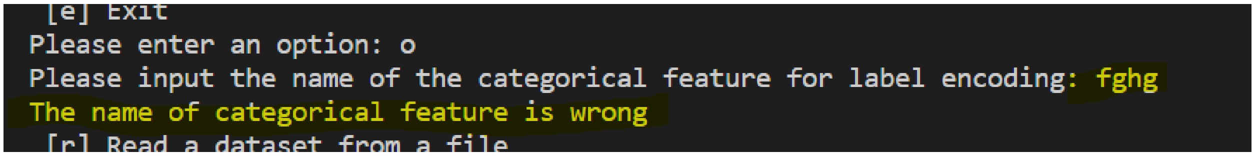


```
id;age;height;weight;active;smoke;gender;governorate;
1;30;170;88;no;yes;1;2;
2;25;160;65;no;no;0;2;
3;28;165;72;yes;yes;1;1;
4;44;188;90;no;no;1;0;
5;60;166;70;no;no;0;0;
```

```
  [s] save the processed dataset
  [e] Exit
Please enter an option: l
Please input the name of the categorical feature for label encoding: gender
The distinct values of the categorical features and their codes are:
female 0
male 1
  [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 scalling
  [s] save the processed dataset
  [e] Exit
Please enter an option: l
Please input the name of the categorical feature for label encoding: governorate
The distinct values of the categorical features and their codes are:
jerusalem 0
nablus 1
ramallah 2
  [r] Read a dataset from a file
```

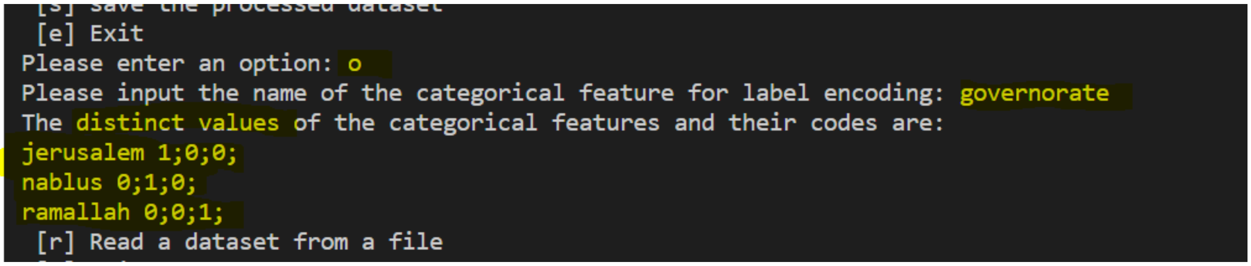
- **For one-hot encoding:**

program verifying that the entered name of the categorical feature exists in the dataset or not :

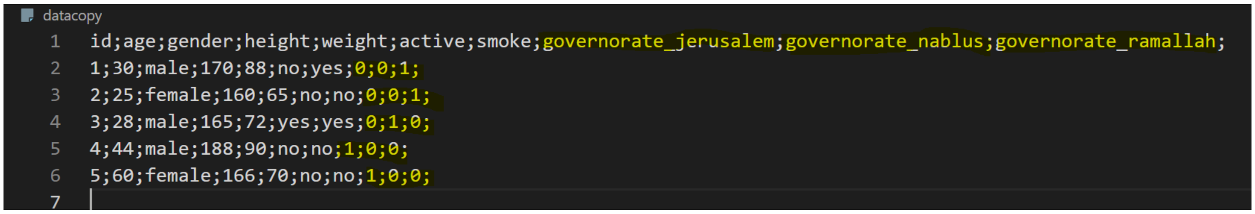


```
  [e] Exit
Please enter an option: o
Please input the name of the categorical feature for label encoding: fghg
The name of categorical feature is wrong
  [r] Read a dataset from a file
```

One-hot encoding for ‘governorate’ feature:

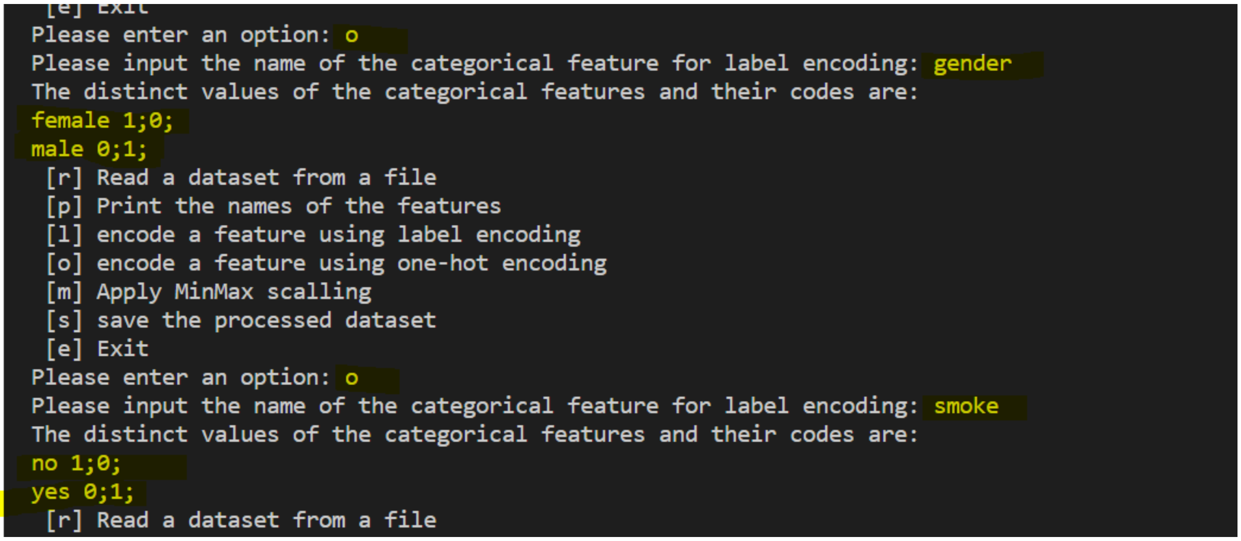


```
 [s] save the processed dataset
 [e] Exit
Please enter an option: o
Please input the name of the categorical feature for label encoding: governorate
The distinct values of the categorical features and their codes are:
jerusalem 1;0;0;
nablus 0;1;0;
ramallah 0;0;1;
 [r] Read a dataset from a file
```

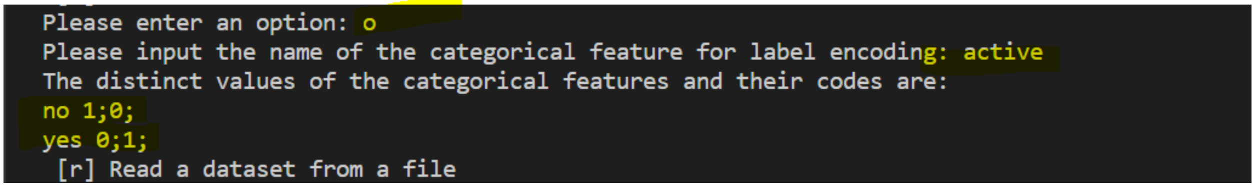


```
id;age;gender;height;weight;active;smoke;governorate_jerusalem;governorate_nablus;governorate_ramallah;
1;30;male;170;88;no;yes;0;0;1;
2;25;female;160;65;no;no;0;0;1;
3;28;male;165;72;yes;yes;0;1;0;
4;44;male;188;90;no;no;1;0;0;
5;60;female;166;70;no;no;1;0;0;
```

Testing one-hot encoding for other features:

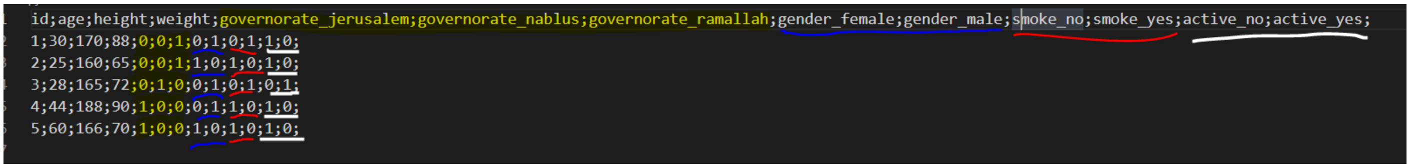


```
 [e] Exit
Please enter an option: o
Please input the name of the categorical feature for label encoding: gender
The distinct values of the categorical features and their codes are:
female 1;0;
male 0;1;
 [r] Read a dataset from a file
 [p] Print the names of the features
 [l] encode a feature using label encoding
 [o] encode a feature using one-hot encoding
 [m] Apply MinMax scalling
 [s] save the processed dataset
 [e] Exit
Please enter an option: o
Please input the name of the categorical feature for label encoding: smoke
The distinct values of the categorical features and their codes are:
no 1;0;
yes 0;1;
 [r] Read a dataset from a file
```



```
Please enter an option: o
Please input the name of the categorical feature for label encoding: active
The distinct values of the categorical features and their codes are:
no 1;0;
yes 0;1;
 [r] Read a dataset from a file
```

<u>After Appling one-hot encoding for 4 features:</u>



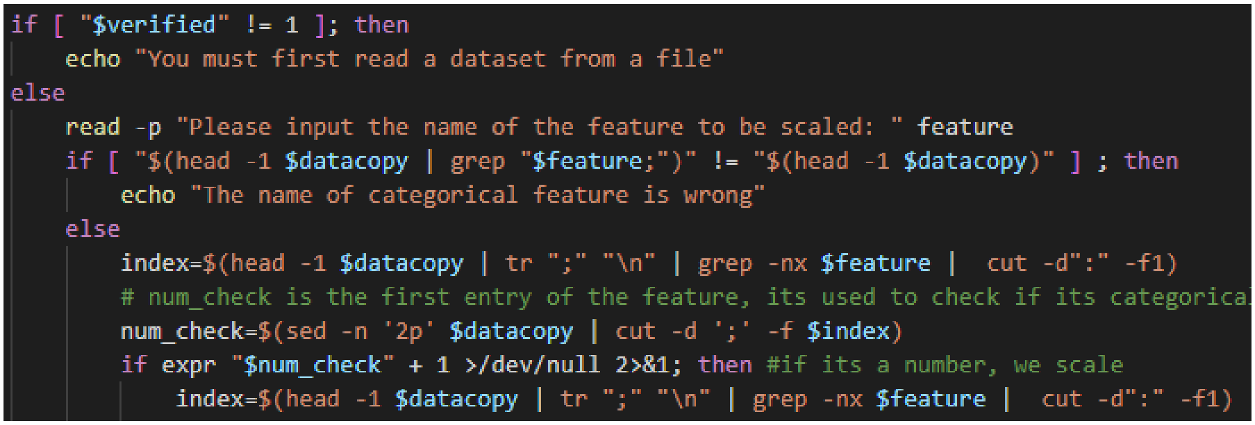
```
id;age;height;weight;governorate_jerusalem;governorate_nablus;governorate_ramallah;gender_female;gender_male;smoke_no;smoke_yes;active_no;active_yes;
1;30;170;88;0;0;1;0;1;0;1;1;0;
2;25;160;65;0;0;1;1;0;1;0;1;0;
3;28;165;72;0;1;0;0;1;0;1;0;1;
4;44;188;90;1;0;0;0;1;1;0;1;0;
5;60;166;70;1;0;0;1;0;1;0;1;0;
```

5. Scaling:

To scale a feature, we do all of the checks we usually do for the previous operations like checking If we read a file already and checking if the feature name is correct. But apart from all of that, we need to check if the feature's values are numeric or not. We do this by finding the value on the 2nd line of that feature and checking if its numeric or not by doing

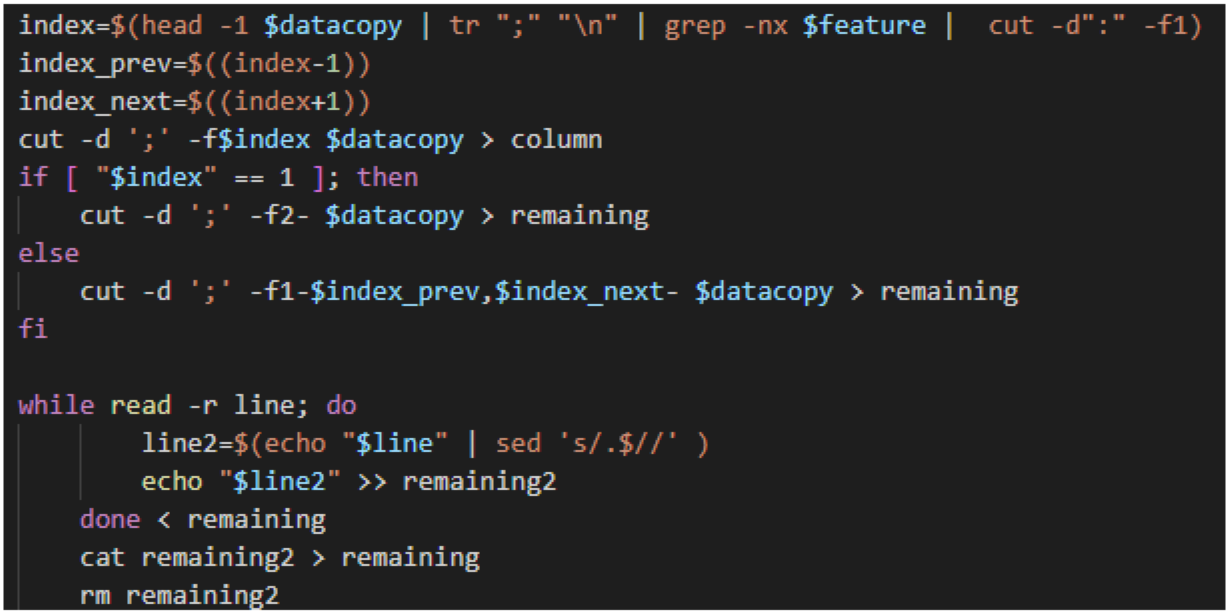
```
expr "$num_check" + 1 >/dev/null 2>&1;
```

If this line returns true, then num_check which is the variable that holds the value on the 2nd line is numeric, otherwise we print an error statement.



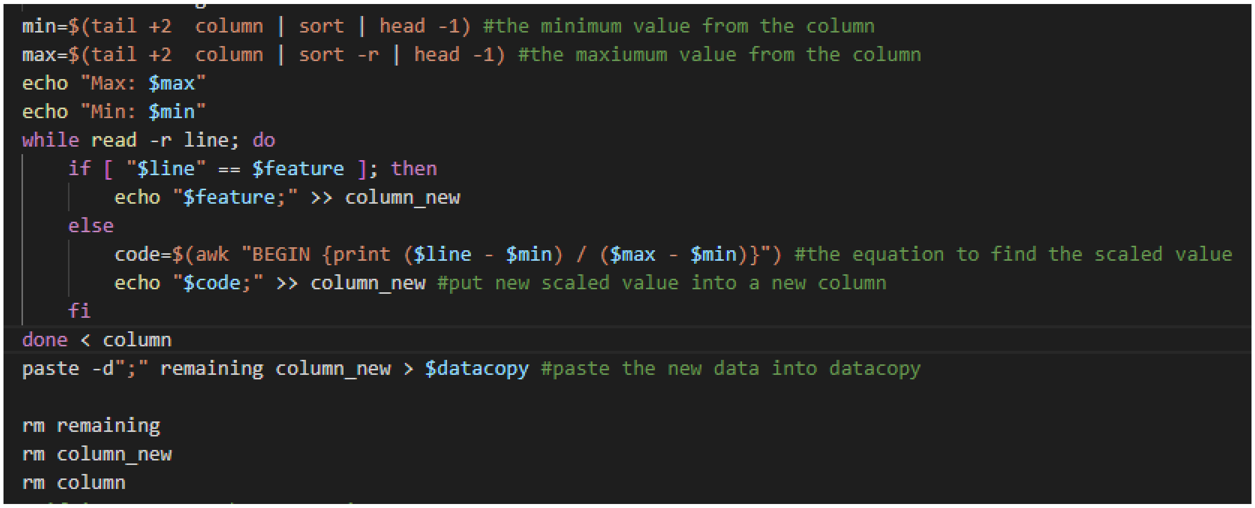
```
if [ "$verified" != 1 ]; then
    echo "You must first read a dataset from a file"
else
    read -p "Please input the name of the feature to be scaled: " feature
    if [ "$(head -1 $datacopy | grep "$feature;")" != "$(head -1 $datacopy)" ] ; then
        echo "The name of categorical feature is wrong"
    else
        index=$(head -1 $datacopy | tr ";" "\n" | grep -nx $feature |  cut -d":" -f1)
        # num_check is the first entry of the feature, its used to check if its categorica
        num_check=$(sed -n '2p' $datacopy | cut -d ';' -f $index)
        if expr "$num_check" + 1 >/dev/null 2>&1; then #if its a number, we scale
            index=$(head -1 $datacopy | tr ";" "\n" | grep -nx $feature |  cut -d":" -f1)
```

Then, we do many things that are similar to what we did before such as creating the column and remaining files using the index.



```
index=$(head -1 $datacopy | tr ";" "\n" | grep -nx $feature |  cut -d":" -f1)
index_prev=$((index-1))
index_next=$((index+1))
cut -d ';' -f$index $datacopy > column
if [ "$index" == 1 ]; then
    cut -d ';' -f2- $datacopy > remaining
else
    cut -d ';' -f1-$index_prev,$index_next- $datacopy > remaining
fi

while read -r line; do
        line2=$(echo "$line" | sed 's/.$//' )
        echo "$line2" >> remaining2
    done < remaining
    cat remaining2 > remaining
    rm remaining2
```

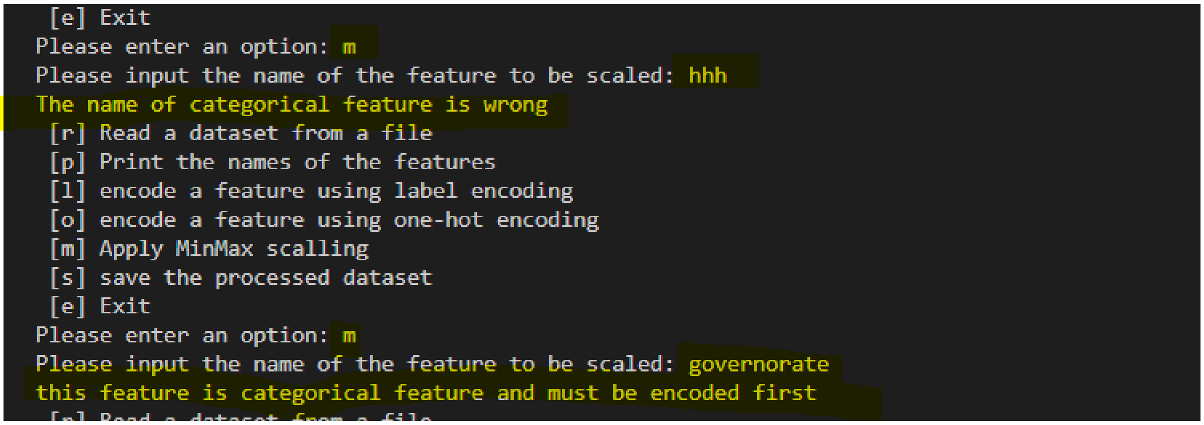
Finally, we find the minimum and maximum values in the column file by removing the first line which contains the feature name using tail and then sorting the file and keeping only the first line using head. Then we use the awk command to find the new scaled code using the equation provided in the project description and add it to a new column then paste the new column with the remaining columns into the datacopy file. At the end, we remove any temporary files we've created.



```
min=$(tail +2  column | sort | head -1) #the minimum value from the column
max=$(tail +2  column | sort -r | head -1) #the maxiumum value from the column
echo "Max: $max"
echo "Min: $min"
while read -r line; do
    if [ "$line" == $feature ]; then
        echo "$feature;" >> column_new
    else
        code=$(awk "BEGIN {print ($line - $min) / ($max - $min)}") #the equation to find the scaled value
        echo "$code;" >> column_new #put new scaled value into a new column
    fi
done < column
paste -d";" remaining column_new > $datacopy #paste the new data into datacopy

rm remaining
rm column_new
rm column
```

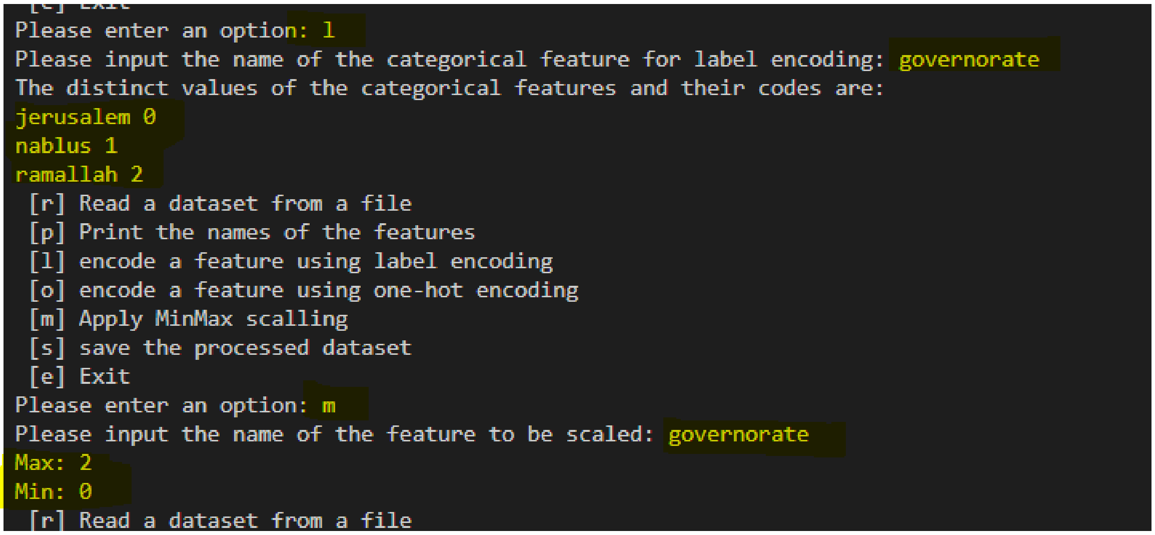
- **Running For scaling :**

Trying to enter wrong feature name and trying to enter a categorical feature:

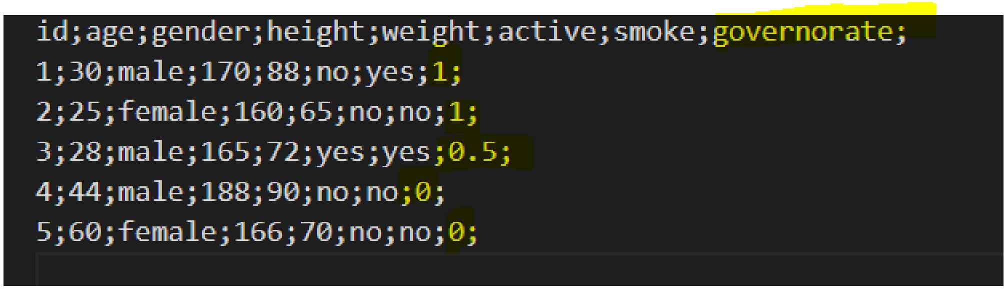


```
 [e] Exit
Please enter an option: m
Please input the name of the feature to be scaled: hhh
The name of categorical feature is wrong
 [r] Read a dataset from a file
 [p] Print the names of the features
 [l] encode a feature using label encoding
 [o] encode a feature using one-hot encoding
 [m] Apply MinMax scalling
 [s] save the processed dataset
 [e] Exit
Please enter an option: m
Please input the name of the feature to be scaled: governorate
this feature is categorical feature and must be encoded first
 [r] Read a dataset from a file
```

Encoding the categorical feature then scaling it:



```
 [e] Exit
Please enter an option: l
Please input the name of the categorical feature for label encoding: governorate
The distinct values of the categorical features and their codes are:
jerusalem 0
nablus 1
ramallah 2
 [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 scalling
 [s] save the processed dataset
 [e] Exit
Please enter an option: m
Please input the name of the feature to be scaled: governorate
Max: 2
Min: 0
 [r] Read a dataset from a file
```



```
id;age;gender;height;weight;active;smoke;governorate;
1;30;male;170;88;no;yes;1;
2;25;female;160;65;no;no;1;
3;28;male;165;72;yes;yes;0.5;
4;44;male;188;90;no;no;0;
5;60;female;166;70;no;no;0;
```

$$x_{i,scaled} = \frac{x_i - x_{min}}{x_{max} - x_{min}}$$

So:

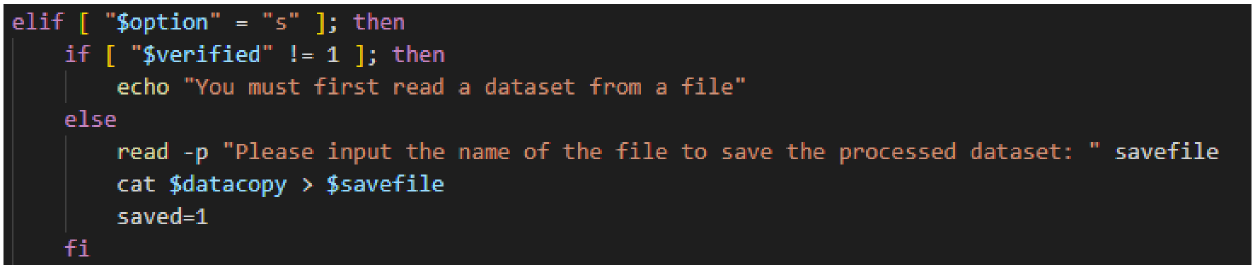
Scaling for Ramallah= 2-0/2-0=1

Scaling for Nablus= 1-0/2-0=0.5

Scaling for Jerusalem= 0-0/2-0=0

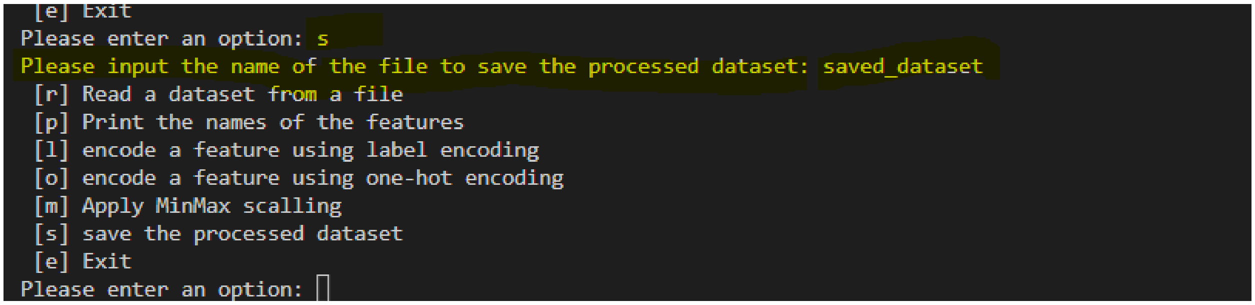
6. Saving:

To save a file, we simple use cat to put the contents of the file into a new file defined by the user, we also set the saved flag to 1.

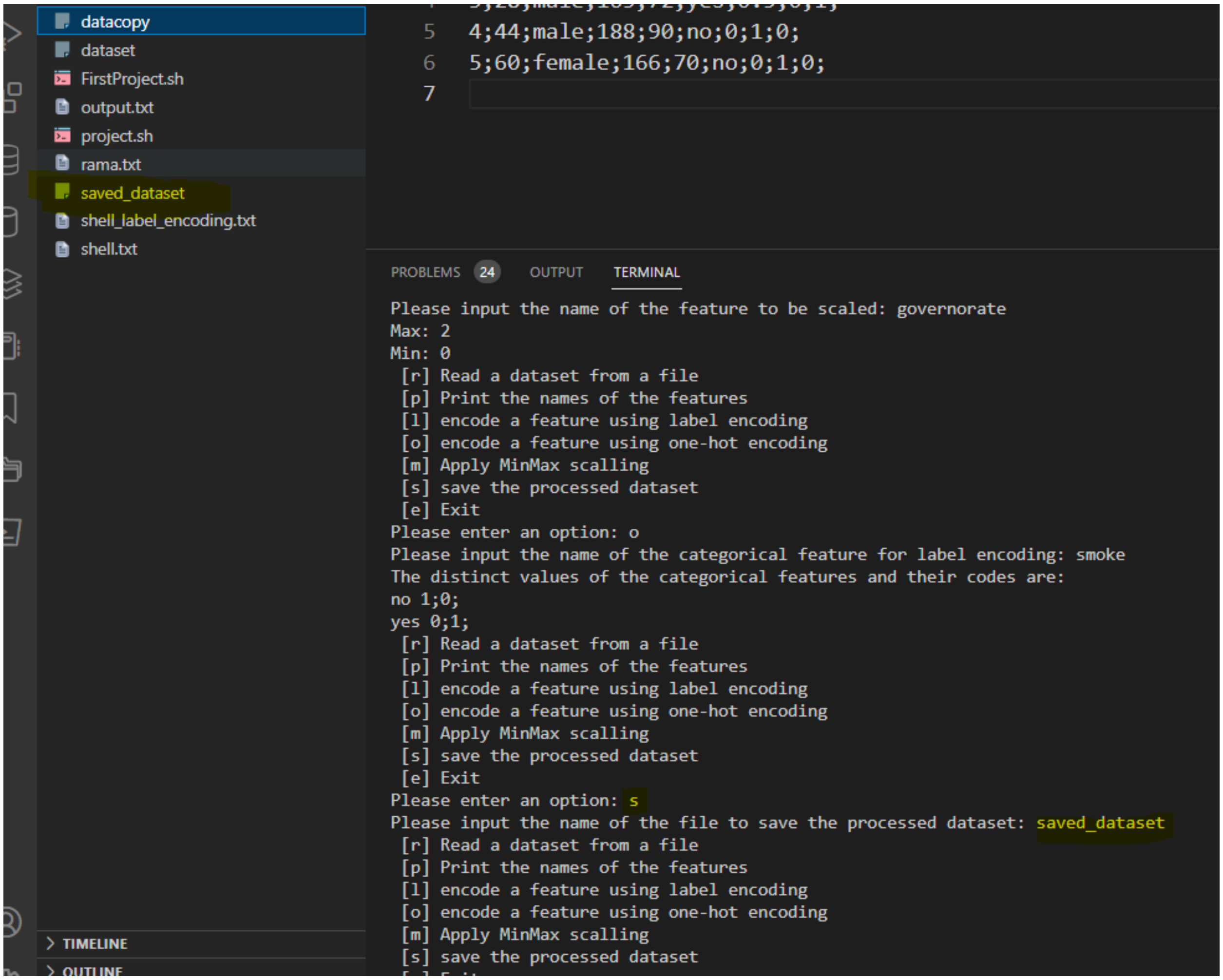


```
elif [ "$option" = "s" ]; then
    if [ "$verified" != 1 ]; then
        echo "You must first read a dataset from a file"
    else
        read -p "Please input the name of the file to save the processed dataset: " savefile
        cat $datacopy > $savefile
        saved=1
    fi
```

- **<u>Running saving:</u>**



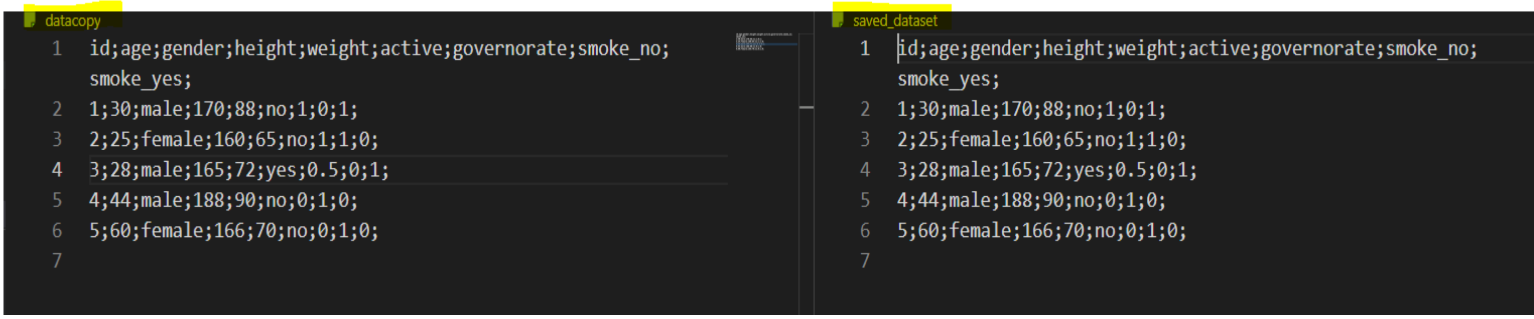
```
  [e] Exit
 Please enter an option: s
 Please input the name of the file to save the processed dataset: saved_dataset
  [r] Read a dataset from a file
  [p] Print the names of the features
  [l] encode a feature using label encoding
  [o] encode a feature using one-hot encoding
  [m] Apply MinMax scalling
  [s] save the processed dataset
  [e] Exit
 Please enter an option: 
```



```
3;28;male;165;72;yes;0.5;0;1;
4;44;male;188;90;no;0;1;0;
5;60;female;166;70;no;0;1;0;
```

```
Please input the name of the feature to be scaled: governorate
Max: 2
Min: 0
 [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 scalling
 [s] save the processed dataset
 [e] Exit
Please enter an option: o
Please input the name of the categorical feature for label encoding: smoke
The distinct values of the categorical features and their codes are:
no 1;0;
yes 0;1;
 [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 scalling
 [s] save the processed dataset
 [e] Exit
Please enter an option: s
Please input the name of the file to save the processed dataset: saved_dataset
 [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 scalling
 [s] save the processed dataset
```

saving 'datacopy' file to 'saved_dataset' file. 'datacopy' file removed after finishing the program, 'saved_dataset' file will be saved in my computer:

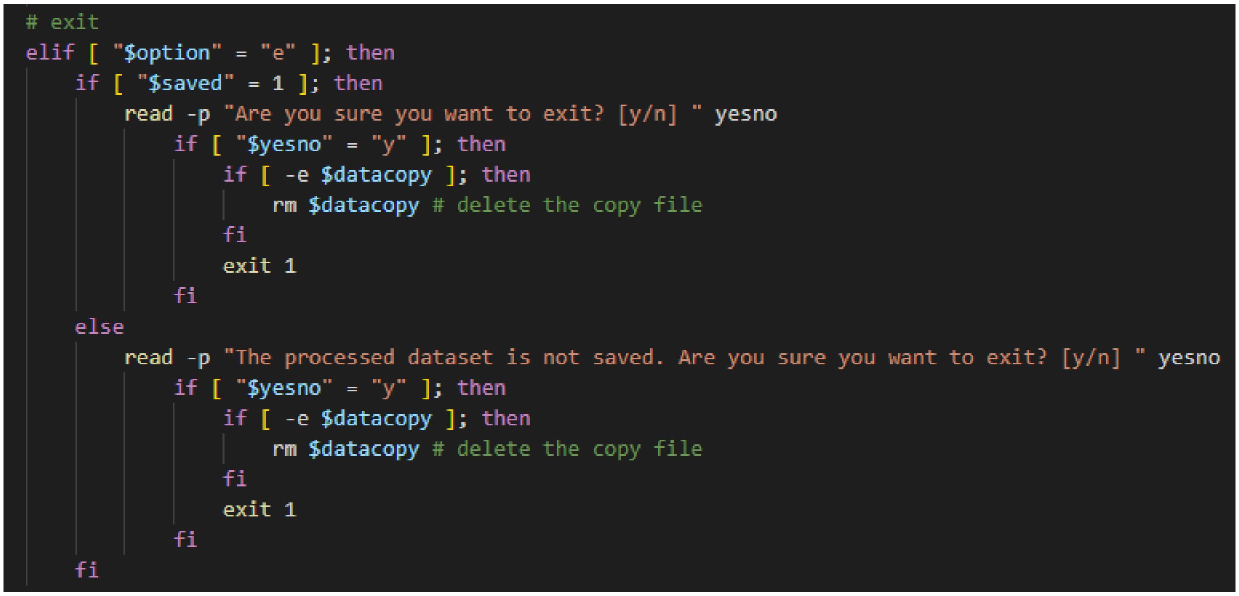


```
id;age;gender;height;weight;active;governorate;smoke_no;
smoke_yes;
1;30;male;170;88;no;1;0;1;
2;25;female;160;65;no;1;1;0;
3;28;male;165;72;yes;0.5;0;1;
4;44;male;188;90;no;0;1;0;
5;60;female;166;70;no;0;1;0;
```

```
id;age;gender;height;weight;active;governorate;smoke_no;
smoke_yes;
1;30;male;170;88;no;1;0;1;
2;25;female;160;65;no;1;1;0;
3;28;male;165;72;yes;0.5;0;1;
4;44;male;188;90;no;0;1;0;
5;60;female;166;70;no;0;1;0;
```

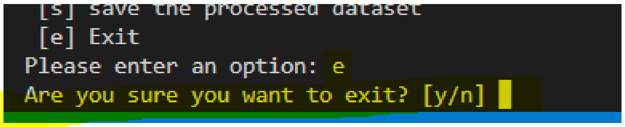
7. Exit:

When the user wants to exit the program, we check if the file has been saved first using the saved flag in order to determine what to print to the user.

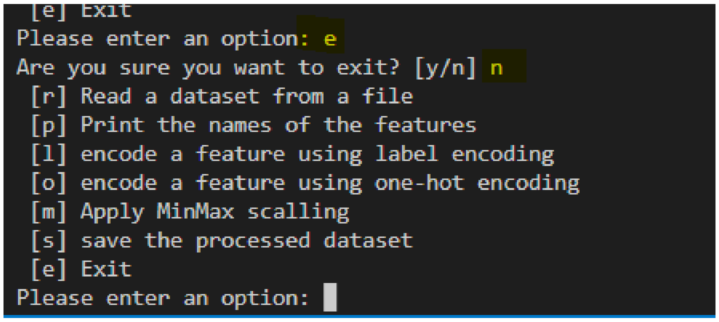


```
# exit
elif [ "$option" = "e" ]; then
    if [ "$saved" = 1 ]; then
        read -p "Are you sure you want to exit? [y/n] " yesno
            if [ "$yesno" = "y" ]; then
                if [ -e $datacopy ]; then
                    rm $datacopy # delete the copy file
                fi
                exit 1
            fi
    else
        read -p "The processed dataset is not saved. Are you sure you want to exit? [y/n] " yesno
            if [ "$yesno" = "y" ]; then
                if [ -e $datacopy ]; then
                    rm $datacopy # delete the copy file
                fi
                exit 1
            fi
    fi
```

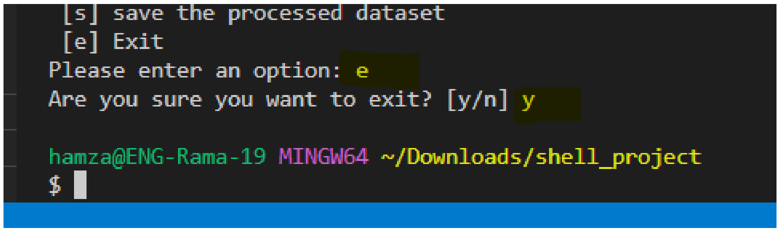
- **Using ‘e’ option:**
  When changes are saved:



```
 [s] save the processed dataset
 [e] Exit
Please enter an option: e
Are you sure you want to exit? [y/n] 
```

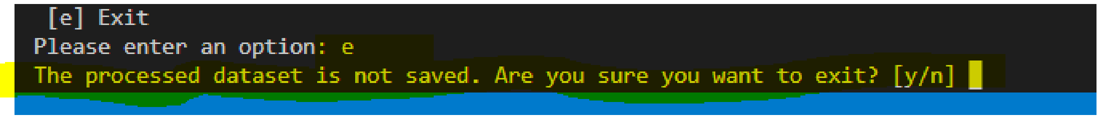


```
 [e] Exit
Please enter an option: e
Are you sure you want to exit? [y/n] n
 [r] Read a dataset from a file
 [p] Print the names of the features
 [l] encode a feature using label encoding
 [o] encode a feature using one-hot encoding
 [m] Apply MinMax scalling
 [s] save the processed dataset
 [e] Exit
Please enter an option: 
```

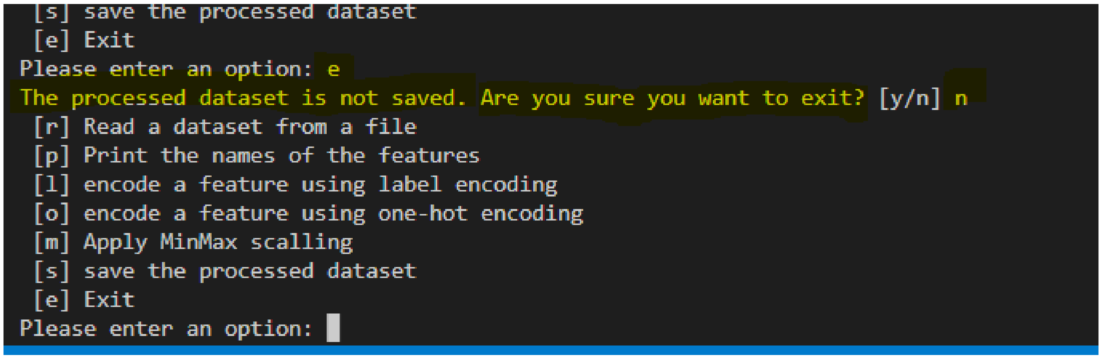


```
  [s] save the processed dataset
  [e] Exit
Please enter an option: e
Are you sure you want to exit? [y/n] y

hamza@ENG-Rama-19 MINGW64 ~/Downloads/shell_project
$ 
```

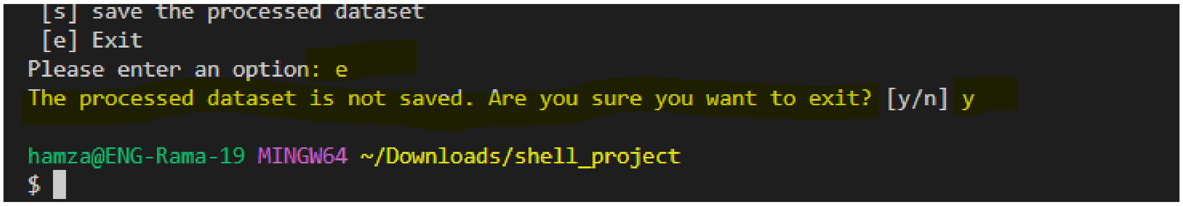
When changes are not saved:



```
 [e] Exit
Please enter an option: e
The processed dataset is not saved. Are you sure you want to exit? [y/n] 
```



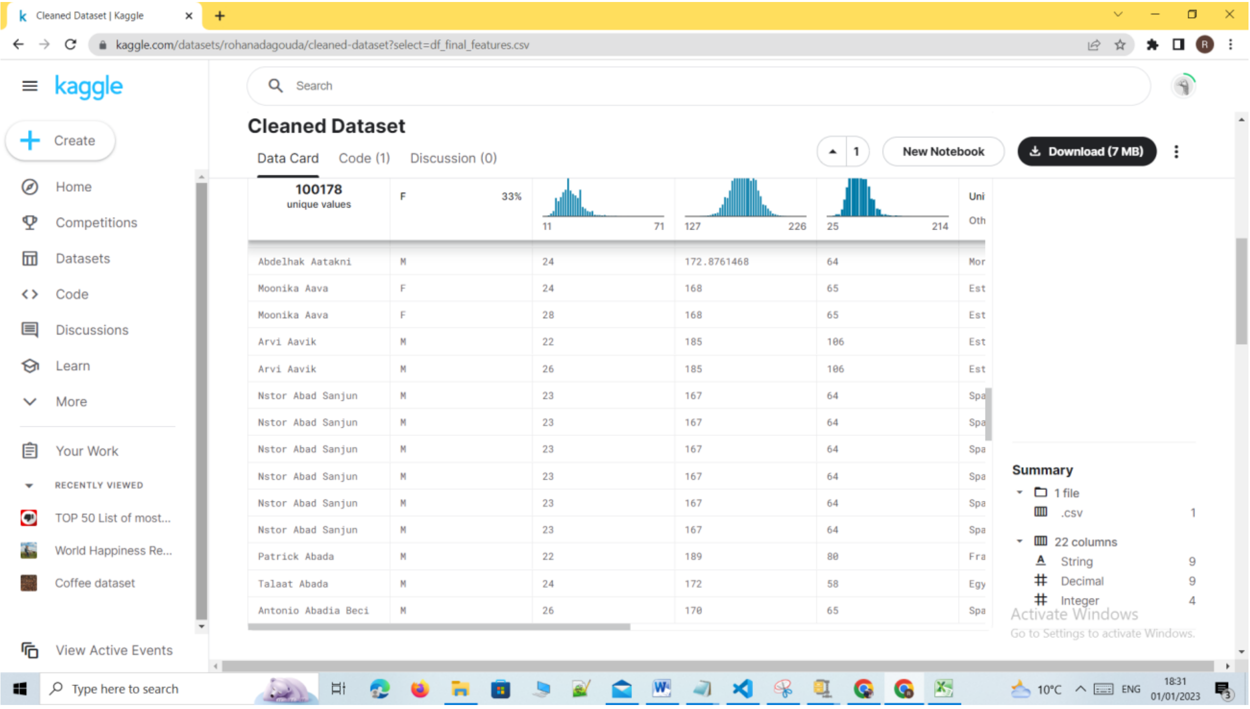
```
  [s] save the processed dataset
  [e] Exit
Please enter an option: e
The processed dataset is not saved. Are you sure you want to exit? [y/n] n
  [r] Read a dataset from a file
  [p] Print the names of the features
  [l] encode a feature using label encoding
  [o] encode a feature using one-hot encoding
  [m] Apply MinMax scalling
  [s] save the processed dataset
  [e] Exit
Please enter an option: 
```



```
  [s] save the processed dataset
  [e] Exit
Please enter an option: e
The processed dataset is not saved. Are you sure you want to exit? [y/n] y

hamza@ENG-Rama-19 MINGW64 ~/Downloads/shell_project
$ 
```

- **Testing for another dataset:**

I have chosen a cleaned dataset from kaggle site:



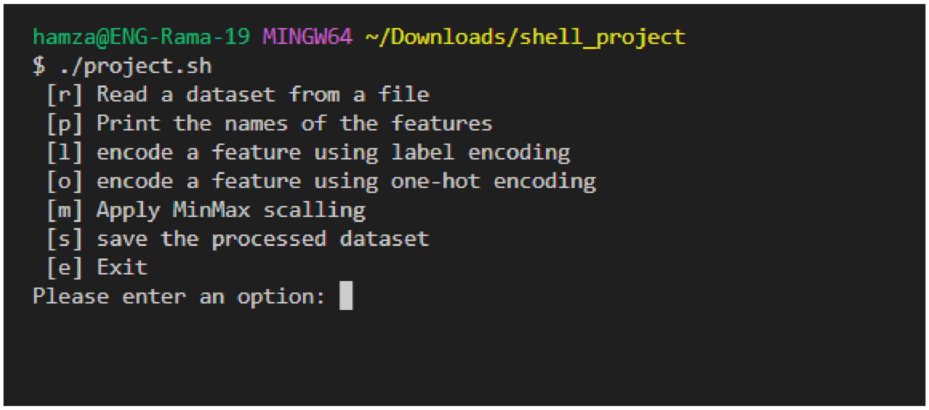
Cleaned Dataset

kaggle.com/datasets/rohanadagouda/cleaned-dataset?select=df_final_features.csv

100178 unique values	F 33%	11 71	127 226	25 214	Uni Oth
Abdelhak Aatakni	M	24	172.8761468	64	Mor
Moonika Aava	F	24	168	65	Est
Moonika Aava	F	28	168	65	Est
Arvi Aavik	M	22	185	106	Est
Arvi Aavik	M	26	185	106	Est
Nstor Abad Sanjun	M	23	167	64	Spa
Nstor Abad Sanjun	M	23	167	64	Spa
Nstor Abad Sanjun	M	23	167	64	Spa
Nstor Abad Sanjun	M	23	167	64	Spa
Nstor Abad Sanjun	M	23	167	64	Spa
Nstor Abad Sanjun	M	23	167	64	Spa
Patrick Abada	M	22	189	80	Fra
Talaat Abada	M	24	172	58	Egy
Antonio Abadia Beci	M	26	170	65	Spa

Summary
1 file
.csv 1
22 columns
String 9
Decimal 9
Integer 4

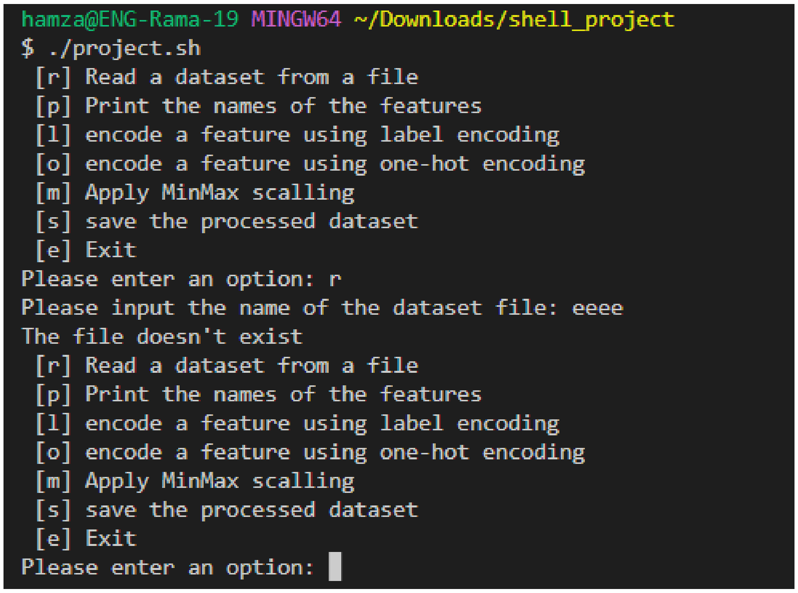
1. **The program should print on the screen the main menu and ask the user to select an option**



```
hamza@ENG-Rama-19 MINGW64 ~/Downloads/shell_project
$ ./project.sh
 [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 scalling
 [s] save the processed dataset
 [e] Exit
Please enter an option: 
```

2. **If the user enters ‘r’:**

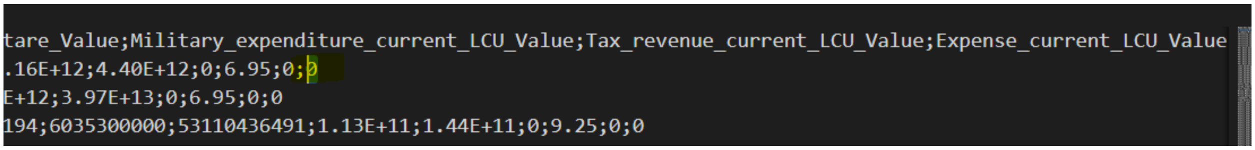
a. The program should print on the screen "Please input the name of the dataset file". As it should verify that the file exists.



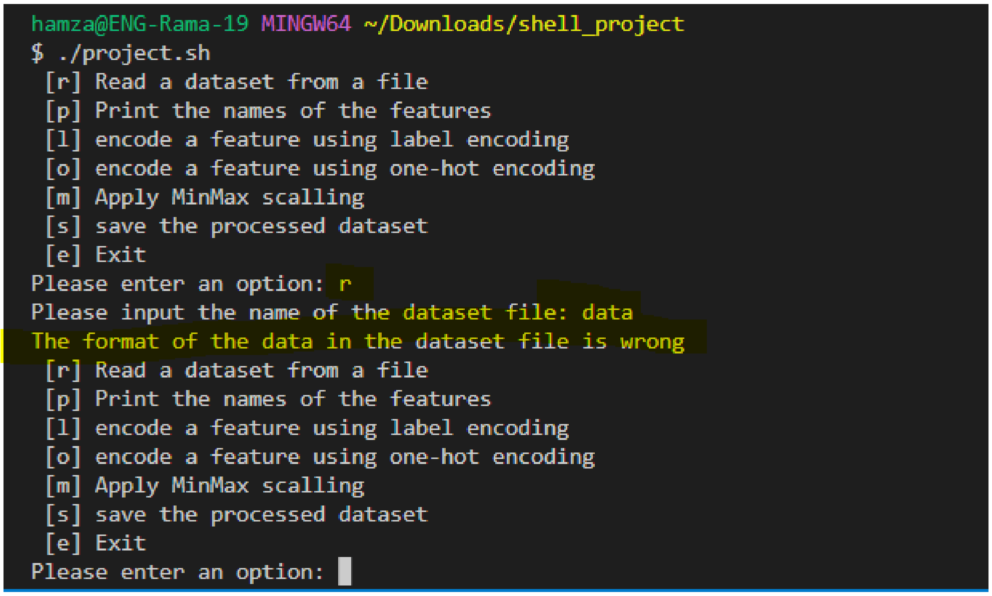
```
hamza@ENG-Rama-19 MINGW64 ~/Downloads/shell_project
$ ./project.sh
 [r] Read a dataset from a file
 [p] Print the names of the features
 [l] encode a feature using label encoding
 [o] encode a feature using one-hot encoding
 [m] Apply MinMax scalling
 [s] save the processed dataset
 [e] Exit
Please enter an option: r
Please input the name of the dataset file: eeee
The file doesn't exist
 [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 scalling
 [s] save the processed dataset
 [e] Exit
Please enter an option: 
```

b. The program should then check the format of the data in the dataset file. In case of any format problems, the program should print on the screen "The format of the data in the dataset file is wrong" and then return to the main menu.

<u>Causing error in format by deleting the value of last column for the first row:</u>

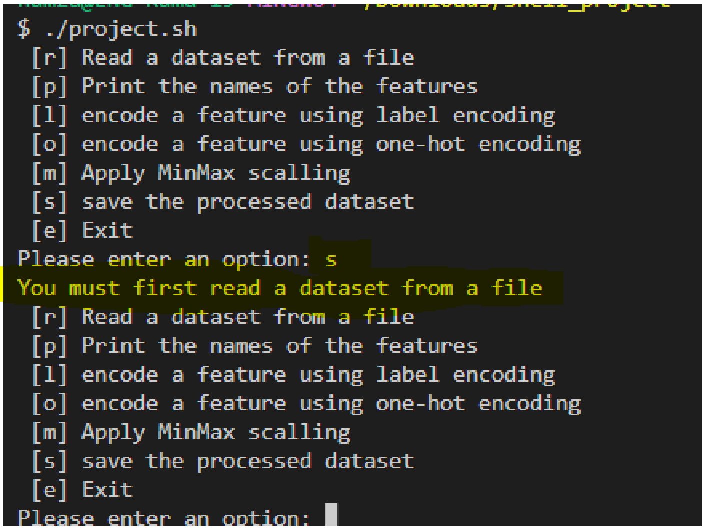


```
tare_Value;Military_expenditure_current_LCU_Value;Tax_revenue_current_LCU_Value;Expense_current_LCU_Value
.16E+12;4.40E+12;0;6.95;0;0
E+12;3.97E+13;0;6.95;0;0
194;6035300000;53110436491;1.13E+11;1.44E+11;0;9.25;0;0
```



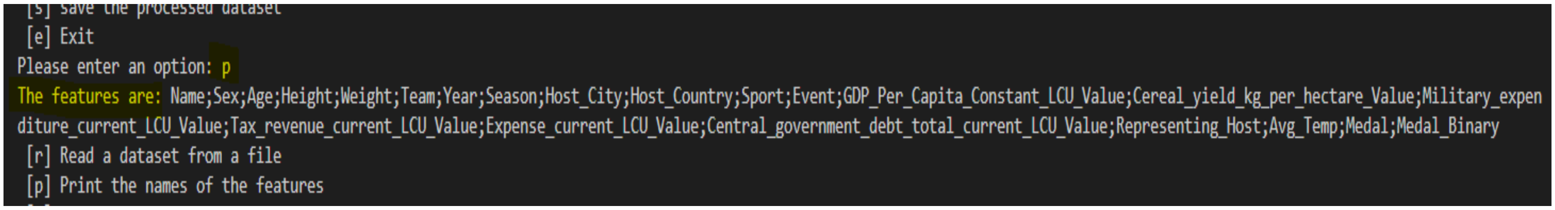
```
hamza@ENG-Rama-19 MINGW64 ~/Downloads/shell_project
$ ./project.sh
 [r] Read a dataset from a file
 [p] Print the names of the features
 [l] encode a feature using label encoding
 [o] encode a feature using one-hot encoding
 [m] Apply MinMax scalling
 [s] save the processed dataset
 [e] Exit
Please enter an option: r
Please input the name of the dataset file: data
The format of the data in the dataset file is wrong
 [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 scalling
 [s] save the processed dataset
 [e] Exit
Please enter an option:
```

c. If the person selects any option other than ‘r’ or ‘e’ before the format of the data in the dataset file is verified correctly, the program should print on the screen “You must first read a dataset from a file” and then return to the main menu.



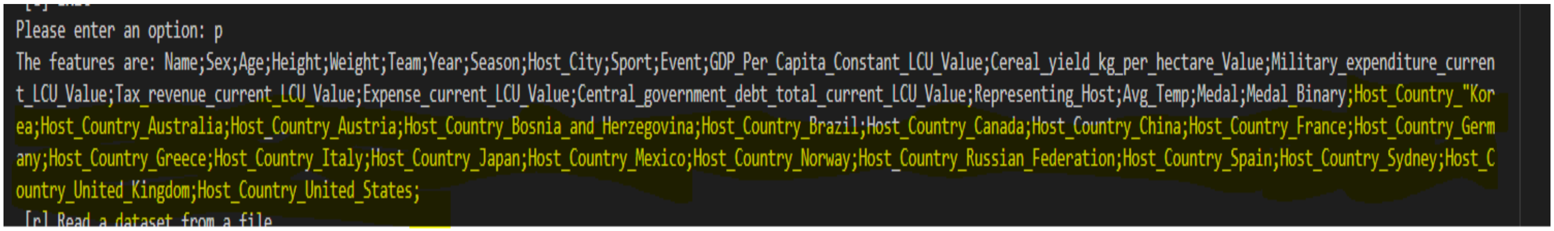
```
hamza@ENG-Rama-19 MINGW64 ~/Downloads/shell_project
$ ./project.sh
 [r] Read a dataset from a file
 [p] Print the names of the features
 [l] encode a feature using label encoding
 [o] encode a feature using one-hot encoding
 [m] Apply MinMax scalling
 [s] save the processed dataset
 [e] Exit
Please enter an option: s
You must first read a dataset from a file
 [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 scalling
 [s] save the processed dataset
 [e] Exit
Please enter an option:
```

3. **If the user enters ‘p’, the program should print on the screen the names of all features of the dataset file and then return to the main menu.**



```
 [s] Save the processed dataset
 [e] Exit
Please enter an option: p
The features are: Name;Sex;Age;Height;Weight;Team;Year;Season;Host_City;Host_Country;Sport;Event;GDP_Per_Capita_Constant_LCU_Value;Cereal_yield_kg_per_hectare_Value;Military_expen
diture_current_LCU_Value;Tax_revenue_current_LCU_Value;Expense_current_LCU_Value;Central_government_debt_total_current_LCU_Value;Representing_Host;Avg_Temp;Medal;Medal_Binary
 [r] Read a dataset from a file
 [p] Print the names of the features
```

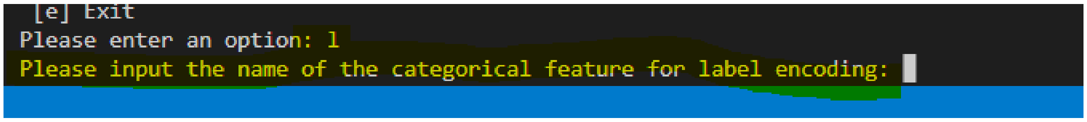
Also printing features after one hot encoding on feature ‘Host_Country':



```
Please enter an option: p
The features are: Name;Sex;Age;Height;Weight;Team;Year;Season;Host_City;Sport;Event;GDP_Per_Capita_Constant_LCU_Value;Cereal_yield_kg_per_hectare_Value;Military_expenditure_curren
t_LCU_Value;Tax_revenue_current_LCU_Value;Expense_current_LCU_Value;Central_government_debt_total_current_LCU_Value;Representing_Host;Avg_Temp;Medal;Medal_Binary;Host_Country_"Kor
ea;Host_Country_Australia;Host_Country_Austria;Host_Country_Bosnia_and_Herzegovina;Host_Country_Brazil;Host_Country_Canada;Host_Country_China;Host_Country_France;Host_Country_Germ
any;Host_Country_Greece;Host_Country_Italy;Host_Country_Japan;Host_Country_Mexico;Host_Country_Norway;Host_Country_Russian_Federation;Host_Country_Spain;Host_Country_Sydney;Host_C
ountry_United_Kingdom;Host_Country_United_States;
 [r] Read a dataset from a file
```

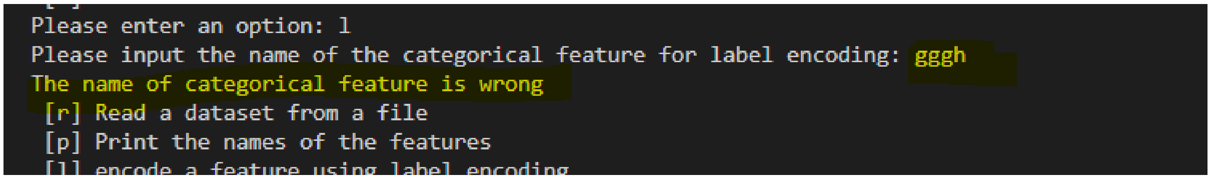
4. If the user enters ‘l’:

a. The program should ask for the name of the feature to be encoded using label encoding by



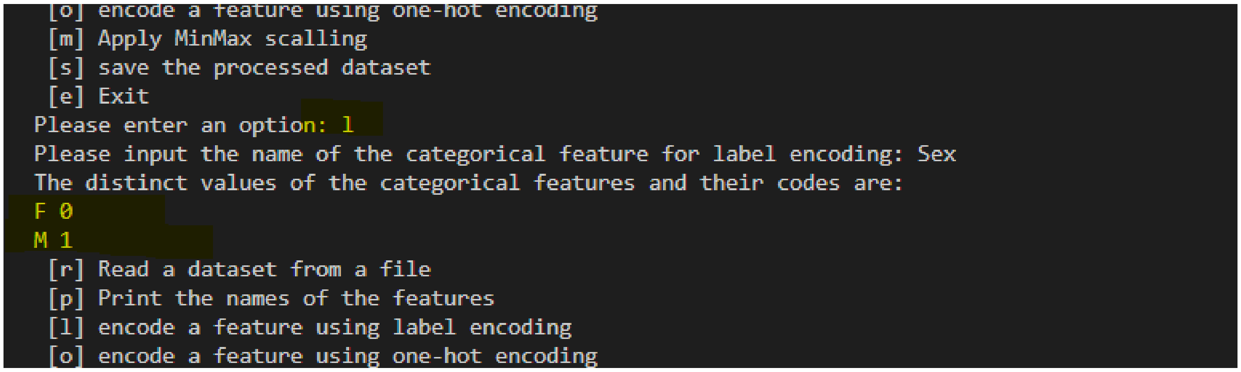
```
   [e] Exit
 Please enter an option: l
 Please input the name of the categorical feature for label encoding: 
```

b. The program should verify if the entered name of the categorical feature exists in the dataset or not.

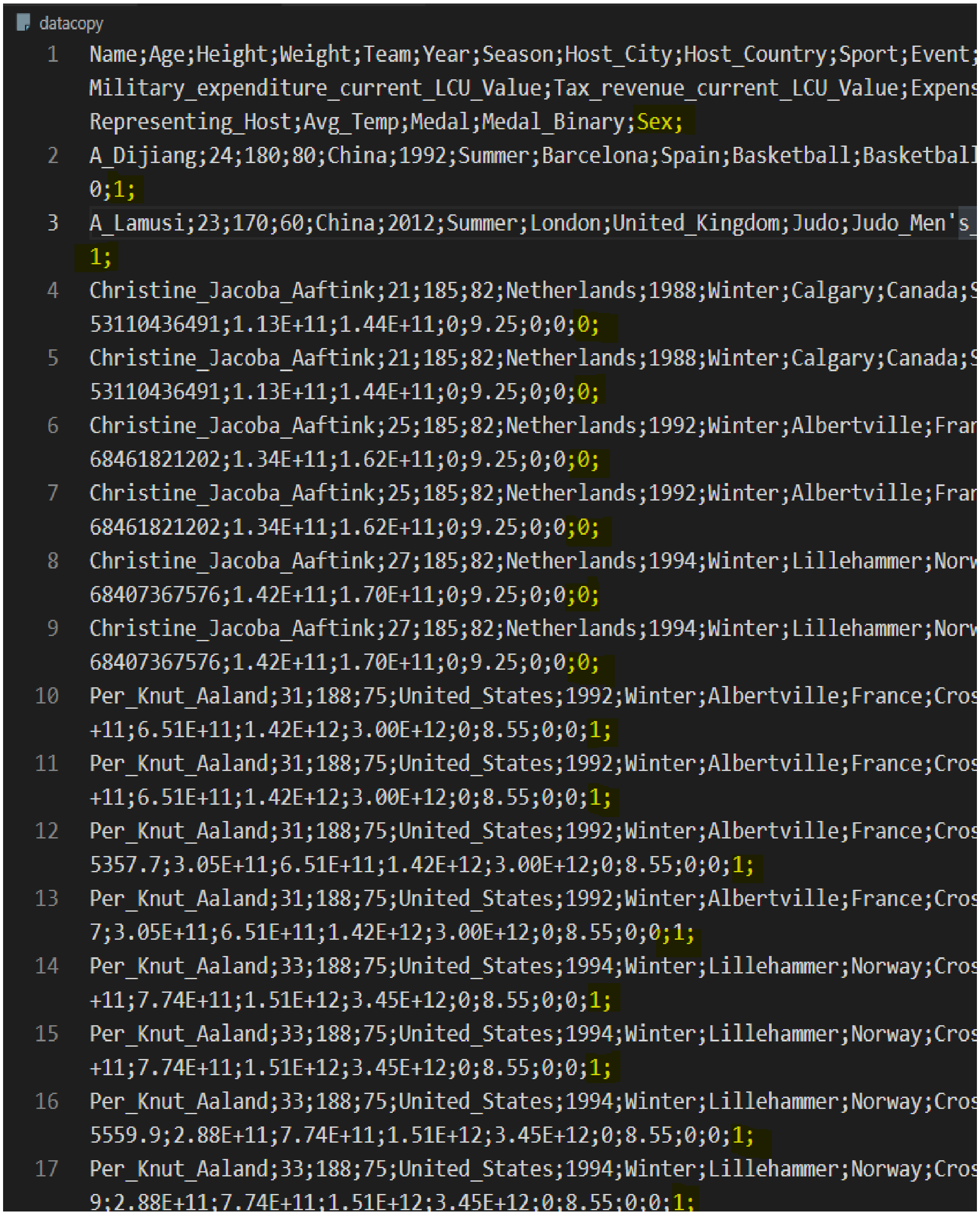


```
 Please enter an option: l
 Please input the name of the categorical feature for label encoding: gggh
 The name of categorical feature is wrong
   [r] Read a dataset from a file
   [p] Print the names of the features
   [l] encode a feature using label encoding
```

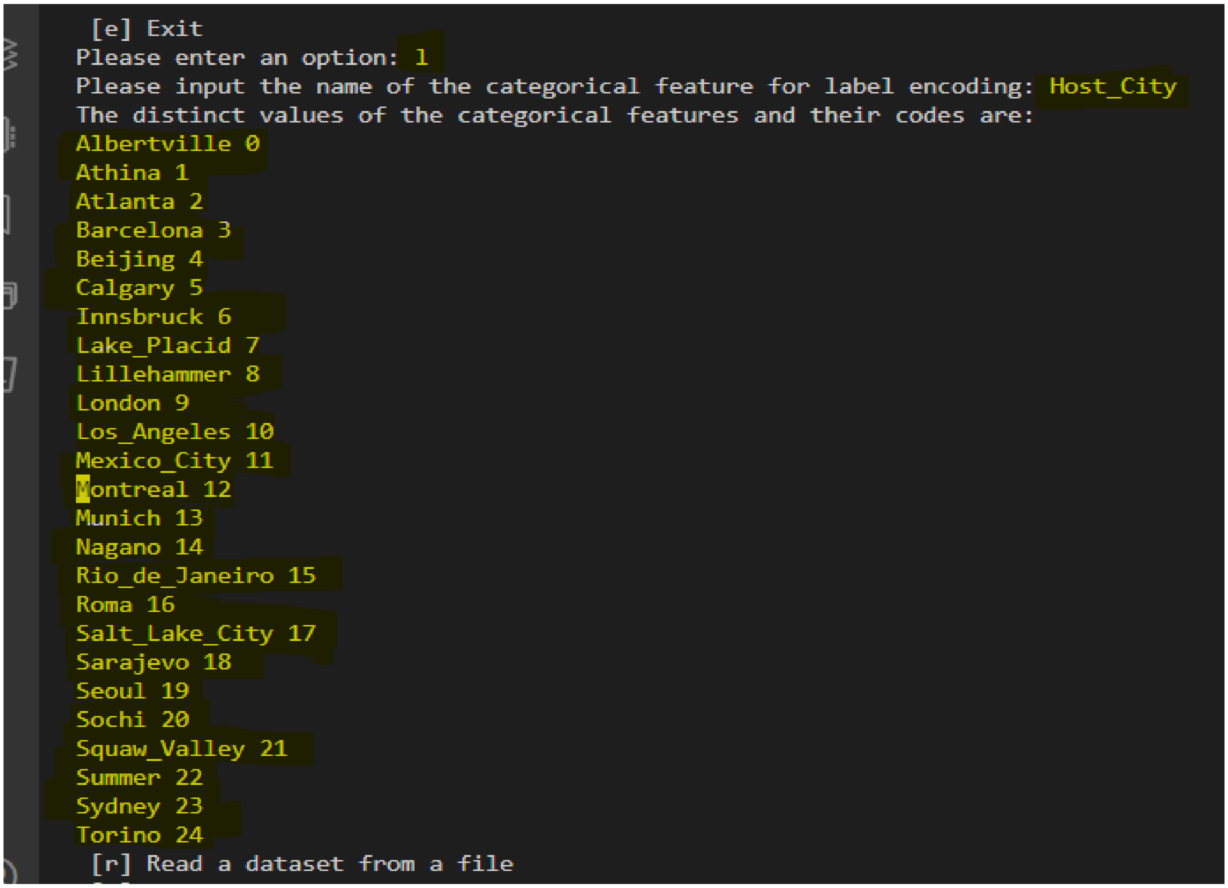
c. If the entered name of the categorical feature exists, the program should print on the screen the distinct values of the categorical feature and the code of each value. And also, to encode the categorical feature in the dataset using label encoding and then return to the main menu.



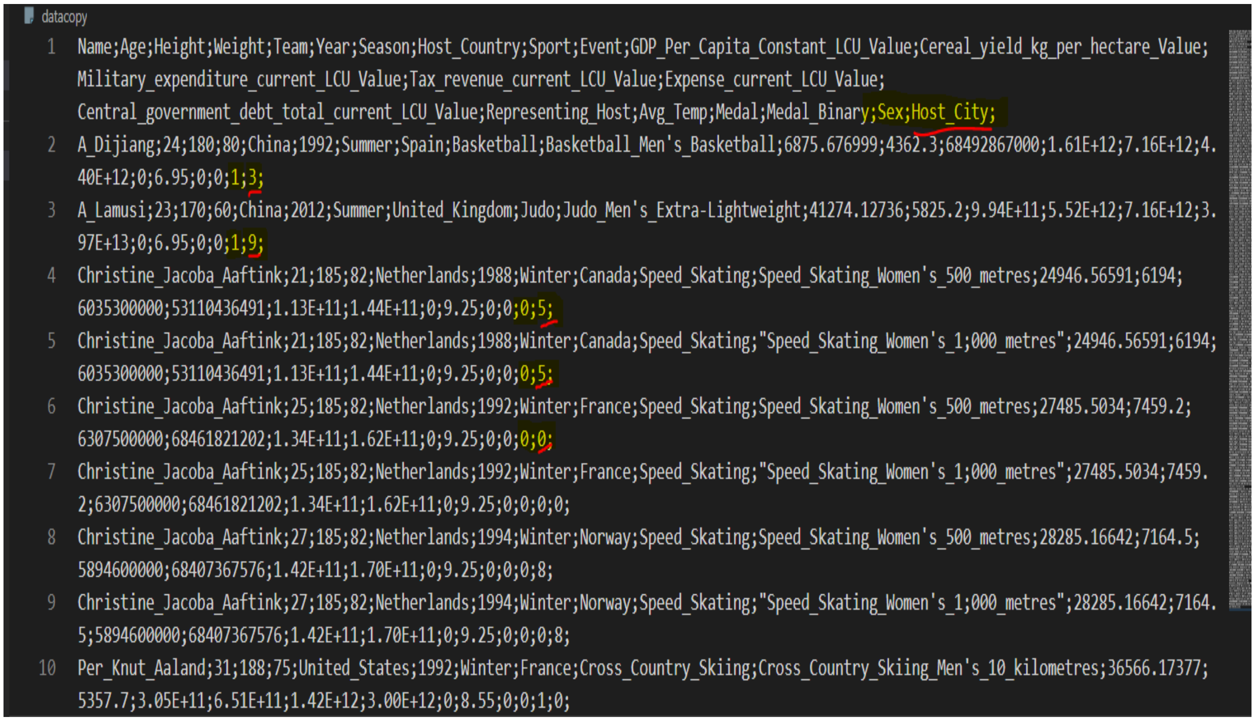
```
   [o] encode a feature using one-hot encoding
   [m] Apply MinMax scalling
   [s] save the processed dataset
   [e] Exit
 Please enter an option: l
 Please input the name of the categorical feature for label encoding: Sex
 The distinct values of the categorical features and their codes are:
 F 0
 M 1
   [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
```



```
Name;Age;Height;Weight;Team;Year;Season;Host_City;Host_Country;Sport;Event;
Military_expenditure_current_LCU_Value;Tax_revenue_current_LCU_Value;Expens
Representing_Host;Avg_Temp;Medal;Medal_Binary;Sex;
A_Dijiang;24;180;80;China;1992;Summer;Barcelona;Spain;Basketball;Basketball
0;1;
A_Lamusi;23;170;60;China;2012;Summer;London;United_Kingdom;Judo;Judo_Men's_
1;
Christine_Jacoba_Aaftink;21;185;82;Netherlands;1988;Winter;Calgary;Canada;S
53110436491;1.13E+11;1.44E+11;0;9.25;0;0;0;
Christine_Jacoba_Aaftink;21;185;82;Netherlands;1988;Winter;Calgary;Canada;S
53110436491;1.13E+11;1.44E+11;0;9.25;0;0;0;
Christine_Jacoba_Aaftink;25;185;82;Netherlands;1992;Winter;Albertville;Fran
68461821202;1.34E+11;1.62E+11;0;9.25;0;0;0;
Christine_Jacoba_Aaftink;25;185;82;Netherlands;1992;Winter;Albertville;Fran
68461821202;1.34E+11;1.62E+11;0;9.25;0;0;0;
Christine_Jacoba_Aaftink;27;185;82;Netherlands;1994;Winter;Lillehammer;Norw
68407367576;1.42E+11;1.70E+11;0;9.25;0;0;0;
Christine_Jacoba_Aaftink;27;185;82;Netherlands;1994;Winter;Lillehammer;Norw
68407367576;1.42E+11;1.70E+11;0;9.25;0;0;0;
Per_Knut_Aaland;31;188;75;United_States;1992;Winter;Albertville;France;Cros
+11;6.51E+11;1.42E+12;3.00E+12;0;8.55;0;0;1;
Per_Knut_Aaland;31;188;75;United_States;1992;Winter;Albertville;France;Cros
+11;6.51E+11;1.42E+12;3.00E+12;0;8.55;0;0;1;
Per_Knut_Aaland;31;188;75;United_States;1992;Winter;Albertville;France;Cros
5357.7;3.05E+11;6.51E+11;1.42E+12;3.00E+12;0;8.55;0;0;1;
Per_Knut_Aaland;31;188;75;United_States;1992;Winter;Albertville;France;Cros
7;3.05E+11;6.51E+11;1.42E+12;3.00E+12;0;8.55;0;0;1;
Per_Knut_Aaland;33;188;75;United_States;1994;Winter;Lillehammer;Norway;Cros
+11;7.74E+11;1.51E+12;3.45E+12;0;8.55;0;0;1;
Per_Knut_Aaland;33;188;75;United_States;1994;Winter;Lillehammer;Norway;Cros
+11;7.74E+11;1.51E+12;3.45E+12;0;8.55;0;0;1;
Per_Knut_Aaland;33;188;75;United_States;1994;Winter;Lillehammer;Norway;Cros
5559.9;2.88E+11;7.74E+11;1.51E+12;3.45E+12;0;8.55;0;0;1;
Per_Knut_Aaland;33;188;75;United_States;1994;Winter;Lillehammer;Norway;Cros
9;2.88E+11;7.74E+11;1.51E+12;3.45E+12;0;8.55;0;0;1;
```



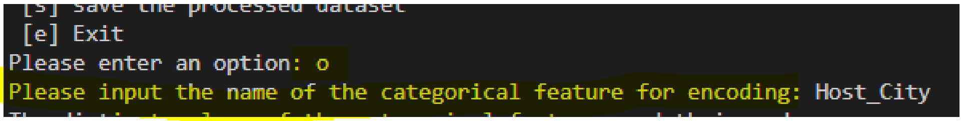
```
  [e] Exit
Please enter an option: 1
Please input the name of the categorical feature for label encoding: Host_City
The distinct values of the categorical features and their codes are:
Albertville 0
Athina 1
Atlanta 2
Barcelona 3
Beijing 4
Calgary 5
Innsbruck 6
Lake_Placid 7
Lillehammer 8
London 9
Los_Angeles 10
Mexico_City 11
Montreal 12
Munich 13
Nagano 14
Rio_de_Janeiro 15
Roma 16
Salt_Lake_City 17
Sarajevo 18
Seoul 19
Sochi 20
Squaw_Valley 21
Summer 22
Sydney 23
Torino 24
  [r] Read a dataset from a file
```



```
Name;Age;Height;Weight;Team;Year;Season;Host_Country;Sport;Event;GDP_Per_Capita_Constant_LCU_Value;Cereal_yield_kg_per_hectare_Value;Military_expenditure_current_LCU_Value;Tax_revenue_current_LCU_Value;Expense_current_LCU_Value;Central_government_debt_total_current_LCU_Value;Representing_Host;Avg_Temp;Medal;Medal_Binary;Sex;Host_City;
A_Dijiang;24;180;80;China;1992;Summer;Spain;Basketball;Basketball_Men's_Basketball;6875.676999;4362.3;68492867000;1.61E+12;7.16E+12;4.40E+12;0;6.95;0;0;1;3;
A_Lamusi;23;170;60;China;2012;Summer;United_Kingdom;Judo;Judo_Men's_Extra-Lightweight;41274.12736;5825.2;9.94E+11;5.52E+12;7.16E+12;3.97E+13;0;6.95;0;0;1;9;
Christine_Jacoba_Aaftink;21;185;82;Netherlands;1988;Winter;Canada;Speed_Skating;Speed_Skating_Women's_500_metres;24946.56591;6194;6035300000;53110436491;1.13E+11;1.44E+11;0;9.25;0;0;0;5;
Christine_Jacoba_Aaftink;21;185;82;Netherlands;1988;Winter;Canada;Speed_Skating;"Speed_Skating_Women's_1;000_metres";24946.56591;6194;6035300000;53110436491;1.13E+11;1.44E+11;0;9.25;0;0;0;5;
Christine_Jacoba_Aaftink;25;185;82;Netherlands;1992;Winter;France;Speed_Skating;Speed_Skating_Women's_500_metres;27485.5034;7459.2;6307500000;68461821202;1.34E+11;1.62E+11;0;9.25;0;0;0;0;
Christine_Jacoba_Aaftink;25;185;82;Netherlands;1992;Winter;France;Speed_Skating;"Speed_Skating_Women's_1;000_metres";27485.5034;7459.2;6307500000;68461821202;1.34E+11;1.62E+11;0;9.25;0;0;0;0;
Christine_Jacoba_Aaftink;27;185;82;Netherlands;1994;Winter;Norway;Speed_Skating;Speed_Skating_Women's_500_metres;28285.16642;7164.5;5894600000;68407367576;1.42E+11;1.70E+11;0;9.25;0;0;0;8;
Christine_Jacoba_Aaftink;27;185;82;Netherlands;1994;Winter;Norway;Speed_Skating;"Speed_Skating_Women's_1;000_metres";28285.16642;7164.5;5894600000;68407367576;1.42E+11;1.70E+11;0;9.25;0;0;0;8;
Per_Knut_Aaland;31;188;75;United_States;1992;Winter;France;Cross_Country_Skiing;Cross_Country_Skiing_Men's_10_kilometres;36566.17377;5357.7;3.05E+11;6.51E+11;1.42E+12;3.00E+12;0;8.55;0;0;1;0;
```

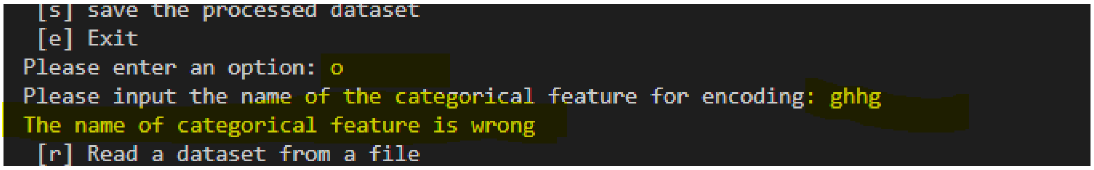
5. **If the user enters ‘o’:**

a. **The program should ask for the name of the feature to be encoded using one-hot encoding by printing on the screen “Please input the name of the categorical feature for one-hot encoding”.**



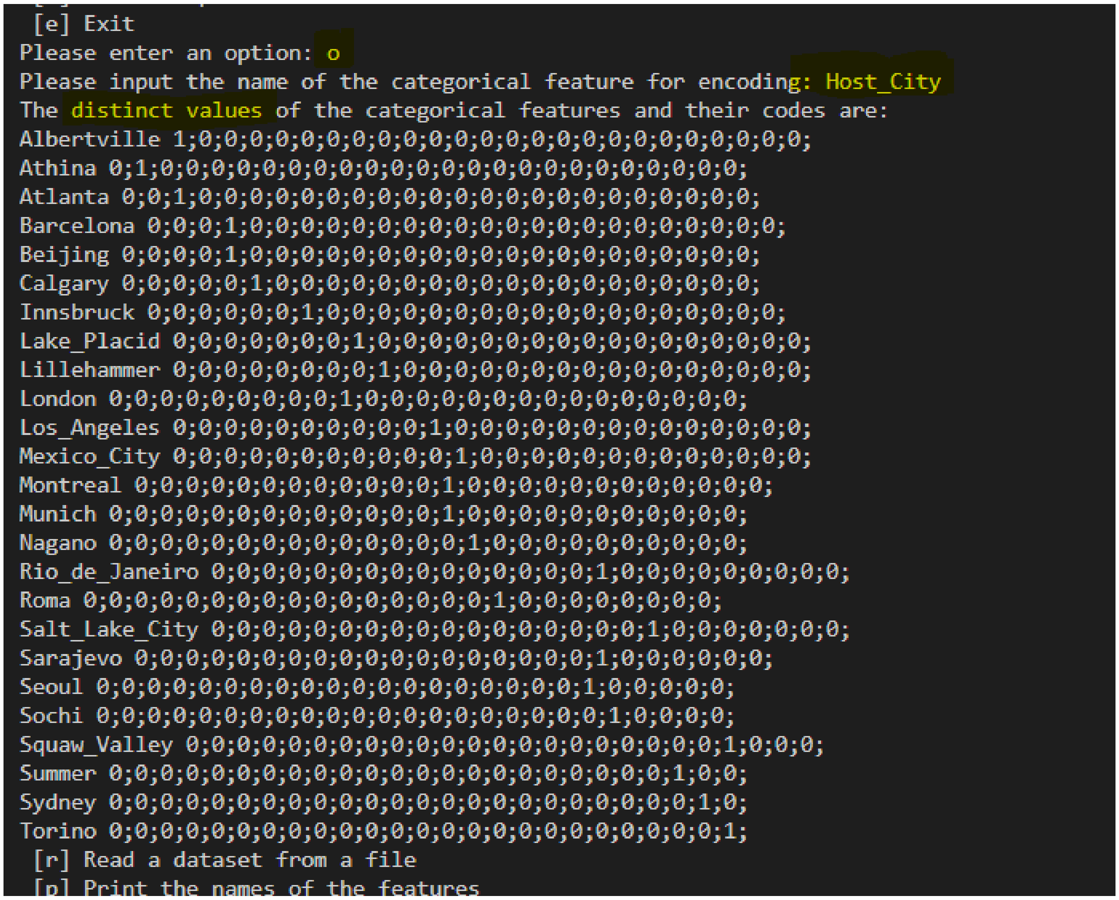
```
  [s] Save the processed dataset
  [e] Exit
 Please enter an option: o
 Please input the name of the categorical feature for encoding: Host_City
```

b. **The program should verify that the entered name of the categorical feature exists in the dataset, or not.**

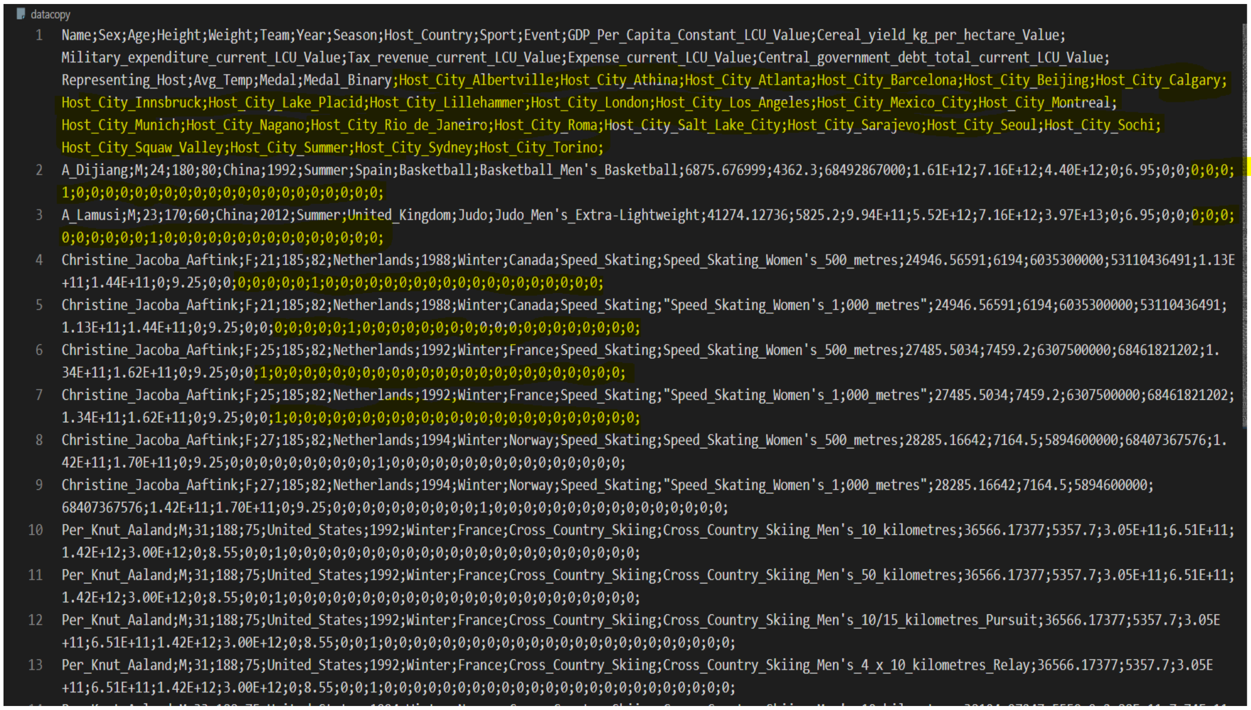


```
  [s] Save the processed dataset
  [e] Exit
 Please enter an option: o
 Please input the name of the categorical feature for encoding: ghhg
 The name of categorical feature is wrong
  [r] Read a dataset from a file
```

c. **If the entered name of the categorical feature exists, the program should then print on the screen the distinct values of the categorical feature. And also, to encode the categorical feature in the dataset using one-hot encoding and then return to the main menu.**



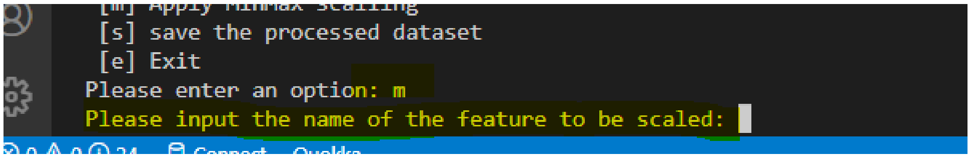
```
 [e] Exit
Please enter an option: o
Please input the name of the categorical feature for encoding: Host_City
The distinct values of the categorical features and their codes are:
Albertville 1;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
Athina 0;1;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
Atlanta 0;0;1;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
Barcelona 0;0;0;1;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
Beijing 0;0;0;0;1;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
Calgary 0;0;0;0;0;1;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
Innsbruck 0;0;0;0;0;0;1;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
Lake_Placid 0;0;0;0;0;0;0;1;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
Lillehammer 0;0;0;0;0;0;0;0;1;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
London 0;0;0;0;0;0;0;0;0;1;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
Los_Angeles 0;0;0;0;0;0;0;0;0;0;1;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
Mexico_City 0;0;0;0;0;0;0;0;0;0;0;1;0;0;0;0;0;0;0;0;0;0;0;0;0;
Montreal 0;0;0;0;0;0;0;0;0;0;0;0;1;0;0;0;0;0;0;0;0;0;0;0;0;
Munich 0;0;0;0;0;0;0;0;0;0;0;0;0;1;0;0;0;0;0;0;0;0;0;0;0;
Nagano 0;0;0;0;0;0;0;0;0;0;0;0;0;0;1;0;0;0;0;0;0;0;0;0;0;
Rio_de_Janeiro 0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;1;0;0;0;0;0;0;0;0;0;
Roma 0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;1;0;0;0;0;0;0;0;0;
Salt_Lake_City 0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;1;0;0;0;0;0;0;0;
Sarajevo 0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;1;0;0;0;0;0;0;
Seoul 0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;1;0;0;0;0;0;
Sochi 0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;1;0;0;0;0;
Squaw_Valley 0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;1;0;0;0;
Summer 0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;1;0;0;
Sydney 0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;1;0;
Torino 0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;1;
 [r] Read a dataset from a file
 [p] Print the names of the features
```



```
datacopy
Name;Sex;Age;Height;Weight;Team;Year;Season;Host_Country;Sport;Event;GDP_Per_Capita_Constant_LCU_Value;Cereal_yield_kg_per_hectare_Value;Military_expenditure_current_LCU_Value;Tax_revenue_current_LCU_Value;Expense_current_LCU_Value;Central_government_debt_total_current_LCU_Value;Representing_Host;Avg_Temp;Medal;Medal_Binary;Host_City_Albertville;Host_City_Athina;Host_City_Atlanta;Host_City_Barcelona;Host_City_Beijing;Host_City_Calgary;Host_City_Innsbruck;Host_City_Lake_Placid;Host_City_Lillehammer;Host_City_London;Host_City_Los_Angeles;Host_City_Mexico_City;Host_City_Montreal;Host_City_Munich;Host_City_Nagano;Host_City_Rio_de_Janeiro;Host_City_Roma;Host_City_Salt_Lake_City;Host_City_Sarajevo;Host_City_Seoul;Host_City_Sochi;Host_City_Squaw_Valley;Host_City_Summer;Host_City_Sydney;Host_City_Torino;
A_Dijiang;M;24;180;80;China;1992;Summer;Spain;Basketball;Basketball_Men's_Basketball;6875.676999;4362.3;68492867000;1.61E+12;7.16E+12;4.40E+12;0;6.95;0;0;0;0;0;1;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
A_Lamusi;M;23;170;60;China;2012;Summer;United_Kingdom;Judo;Judo_Men's_Extra-Lightweight;41274.12736;5825.2;9.94E+11;5.52E+12;7.16E+12;3.97E+13;0;6.95;0;0;0;0;0;0;0;0;0;0;0;1;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
Christine_Jacoba_Aaftink;F;21;185;82;Netherlands;1988;Winter;Canada;Speed_Skating;Speed_Skating_Women's_500_metres;24946.56591;6194;6035300000;53110436491;1.13E+11;1.44E+11;0;9.25;0;0;0;0;0;0;0;1;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
Christine_Jacoba_Aaftink;F;21;185;82;Netherlands;1988;Winter;Canada;Speed_Skating;"Speed_Skating_Women's_1;000_metres";24946.56591;6194;6035300000;53110436491;1.13E+11;1.44E+11;0;9.25;0;0;0;0;0;0;0;1;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
Christine_Jacoba_Aaftink;F;25;185;82;Netherlands;1992;Winter;France;Speed_Skating;Speed_Skating_Women's_500_metres;27485.5034;7459.2;6307500000;68461821202;1.34E+11;1.62E+11;0;9.25;0;0;1;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
Christine_Jacoba_Aaftink;F;25;185;82;Netherlands;1992;Winter;France;Speed_Skating;"Speed_Skating_Women's_1;000_metres";27485.5034;7459.2;6307500000;68461821202;1.34E+11;1.62E+11;0;9.25;0;0;1;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
Christine_Jacoba_Aaftink;F;27;185;82;Netherlands;1994;Winter;Norway;Speed_Skating;Speed_Skating_Women's_500_metres;28285.16642;7164.5;5894600000;68407367576;1.42E+11;1.70E+11;0;9.25;0;0;0;0;0;0;0;0;0;0;1;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
Christine_Jacoba_Aaftink;F;27;185;82;Netherlands;1994;Winter;Norway;Speed_Skating;"Speed_Skating_Women's_1;000_metres";28285.16642;7164.5;5894600000;68407367576;1.42E+11;1.70E+11;0;9.25;0;0;0;0;0;0;0;0;0;0;1;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
Per_Knut_Aaland;M;31;188;75;United_States;1992;Winter;France;Cross_Country_Skiing;Cross_Country_Skiing_Men's_10_kilometres;36566.17377;5357.7;3.05E+11;6.51E+11;1.42E+12;3.00E+12;0;8.55;0;0;1;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
Per_Knut_Aaland;M;31;188;75;United_States;1992;Winter;France;Cross_Country_Skiing;Cross_Country_Skiing_Men's_50_kilometres;36566.17377;5357.7;3.05E+11;6.51E+11;1.42E+12;3.00E+12;0;8.55;0;0;1;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
Per_Knut_Aaland;M;31;188;75;United_States;1992;Winter;France;Cross_Country_Skiing;Cross_Country_Skiing_Men's_10/15_kilometres_Pursuit;36566.17377;5357.7;3.05E+11;6.51E+11;1.42E+12;3.00E+12;0;8.55;0;0;1;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
Per_Knut_Aaland;M;31;188;75;United_States;1992;Winter;France;Cross_Country_Skiing;Cross_Country_Skiing_Men's_4_x_10_kilometres_Relay;36566.17377;5357.7;3.05E+11;6.51E+11;1.42E+12;3.00E+12;0;8.55;0;0;1;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
```

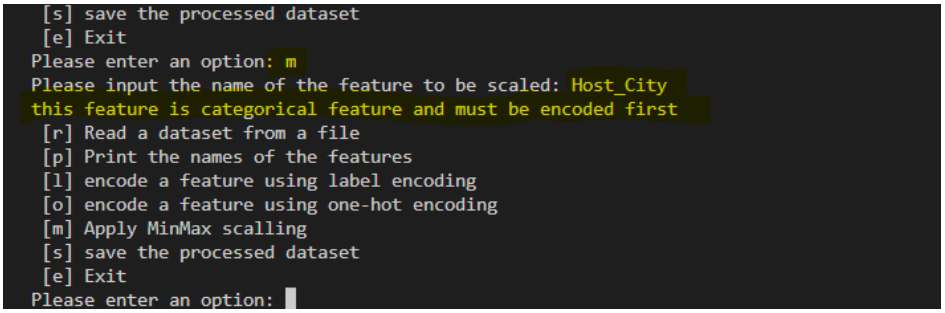
6. **If the user enters ‘m’:**

a. The program should ask for the name of the feature to be scaled using MinMax scaling by printing on the screen “Please input the name of the feature to be scaled”.



```
     [m] Apply MinMax scalling
     [s] save the processed dataset
     [e] Exit
    Please enter an option: m
    Please input the name of the feature to be scaled: 
```

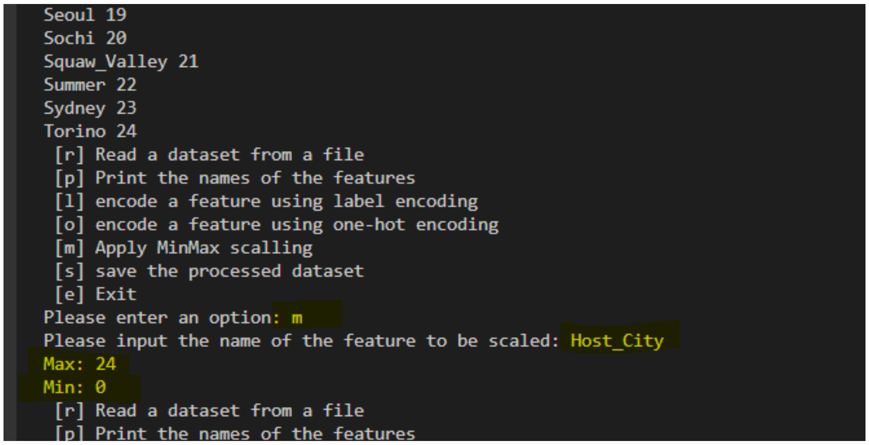
b. If the entered feature is a categorical feature, the program should verify that this feature is encoded, otherwise, the program should print on screen “this feature is categorical feature and must be encoded first” and then return to the main menu.



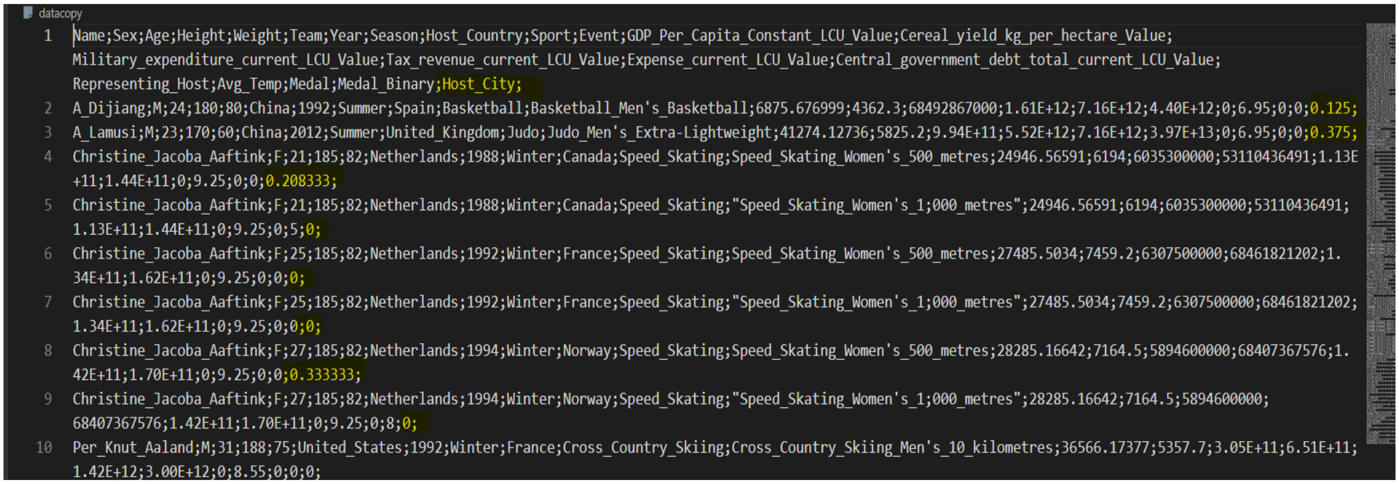
```
 [s] save the processed dataset
 [e] Exit
Please enter an option: m
Please input the name of the feature to be scaled: Host_City
this feature is categorical feature and must be encoded first
 [r] Read a dataset from a file
 [p] Print the names of the features
 [l] encode a feature using label encoding
 [o] encode a feature using one-hot encoding
 [m] Apply MinMax scalling
 [s] save the processed dataset
 [e] Exit
Please enter an option: 
```

d. **If the feature is numeric or encoded categorical feature, the program should print on the screen the minimum and maximum values of the feature and apply the Min-Max scaling to the feature vector and then return to main menu.**

After applying label encoding on ‘Host_City’ feature, We were able to use scaling on this feature, as following:



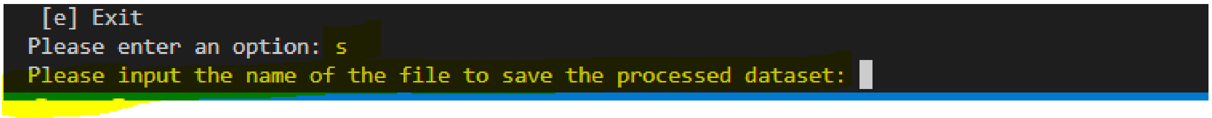
```
Seoul 19
Sochi 20
Squaw_Valley 21
Summer 22
Sydney 23
Torino 24
 [r] Read a dataset from a file
 [p] Print the names of the features
 [l] encode a feature using label encoding
 [o] encode a feature using one-hot encoding
 [m] Apply MinMax scalling
 [s] save the processed dataset
 [e] Exit
Please enter an option: m
Please input the name of the feature to be scaled: Host_City
Max: 24
Min: 0
 [r] Read a dataset from a file
 [p] Print the names of the features
```



```
Name;Sex;Age;Height;Weight;Team;Year;Season;Host_Country;Sport;Event;GDP_Per_Capita_Constant_LCU_Value;Cereal_yield_kg_per_hectare_Value;
Military_expenditure_current_LCU_Value;Tax_revenue_current_LCU_Value;Expense_current_LCU_Value;Central_government_debt_total_current_LCU_Value;
Representing_Host;Avg_Temp;Medal;Medal_Binary;Host_City;
A_Dijiang;M;24;180;80;China;1992;Summer;Spain;Basketball;Basketball_Men's_Basketball;6875.676999;4362.3;68492867000;1.61E+12;7.16E+12;4.40E+12;0;6.95;0;0;0.125;
A_Lamusi;M;23;170;60;China;2012;Summer;United_Kingdom;Judo;Judo_Men's_Extra-Lightweight;41274.12736;5825.2;9.94E+11;5.52E+12;7.16E+12;3.97E+13;0;6.95;0;0;0.375;
Christine_Jacoba_Aaftink;F;21;185;82;Netherlands;1988;Winter;Canada;Speed_Skating;Speed_Skating_Women's_500_metres;24946.56591;6194;6035300000;53110436491;1.13E
+11;1.44E+11;0;9.25;0;0;0.208333;
Christine_Jacoba_Aaftink;F;21;185;82;Netherlands;1988;Winter;Canada;Speed_Skating;"Speed_Skating_Women's_1;000_metres";24946.56591;6194;6035300000;53110436491;
1.13E+11;1.44E+11;0;9.25;0;5;0;
Christine_Jacoba_Aaftink;F;25;185;82;Netherlands;1992;Winter;France;Speed_Skating;Speed_Skating_Women's_500_metres;27485.5034;7459.2;6307500000;68461821202;1.
34E+11;1.62E+11;0;9.25;0;0;0;
Christine_Jacoba_Aaftink;F;25;185;82;Netherlands;1992;Winter;France;Speed_Skating;"Speed_Skating_Women's_1;000_metres";27485.5034;7459.2;6307500000;68461821202;
1.34E+11;1.62E+11;0;9.25;0;0;0;
Christine_Jacoba_Aaftink;F;27;185;82;Netherlands;1994;Winter;Norway;Speed_Skating;Speed_Skating_Women's_500_metres;28285.16642;7164.5;5894600000;68407367576;1.
42E+11;1.70E+11;0;9.25;0;0;0.333333;
Christine_Jacoba_Aaftink;F;27;185;82;Netherlands;1994;Winter;Norway;Speed_Skating;"Speed_Skating_Women's_1;000_metres";28285.16642;7164.5;5894600000;
68407367576;1.42E+11;1.70E+11;0;9.25;0;8;0;
Per_Knut_Aaland;M;31;188;75;United_States;1992;Winter;France;Cross_Country_Skiing;Cross_Country_Skiing_Men's_10_kilometres;36566.17377;5357.7;3.05E+11;6.51E+11;
1.42E+12;3.00E+12;0;8.55;0;0;0;
```

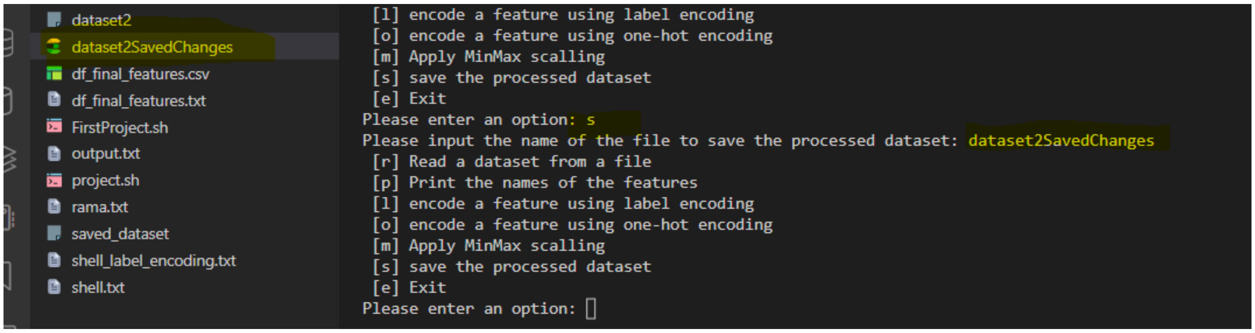
7. **If the user enters 's':**

a. The program should print on the screen "Please input the name of the file to save the processed dataset".



```
 [e] Exit
Please enter an option: s
Please input the name of the file to save the processed dataset: 
```

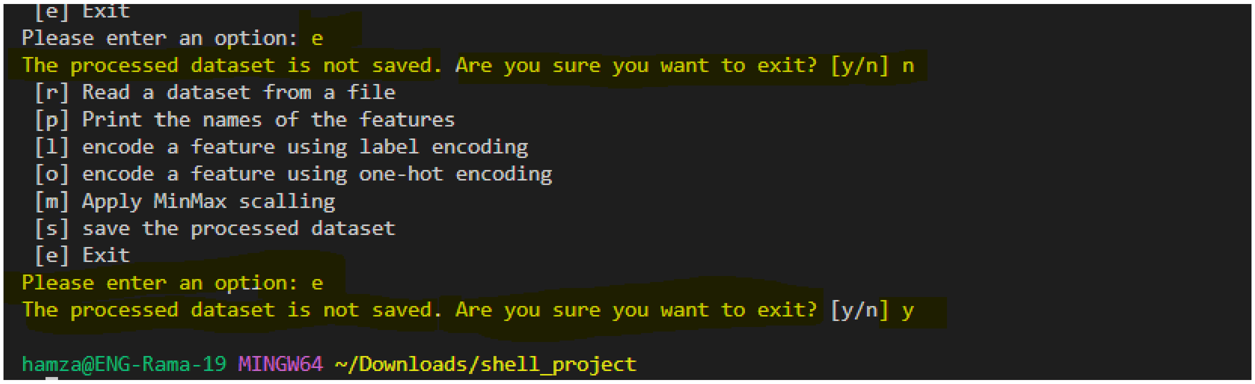
b. The program should save the processed dataset into the entered filename and then return to the main menu.



```
 [l] encode a feature using label encoding
 [o] encode a feature using one-hot encoding
 [m] Apply MinMax scalling
 [s] save the processed dataset
 [e] Exit
Please enter an option: s
Please input the name of the file to save the processed dataset: dataset2SavedChanges
 [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 scalling
 [s] save the processed dataset
 [e] Exit
Please enter an option: 
```

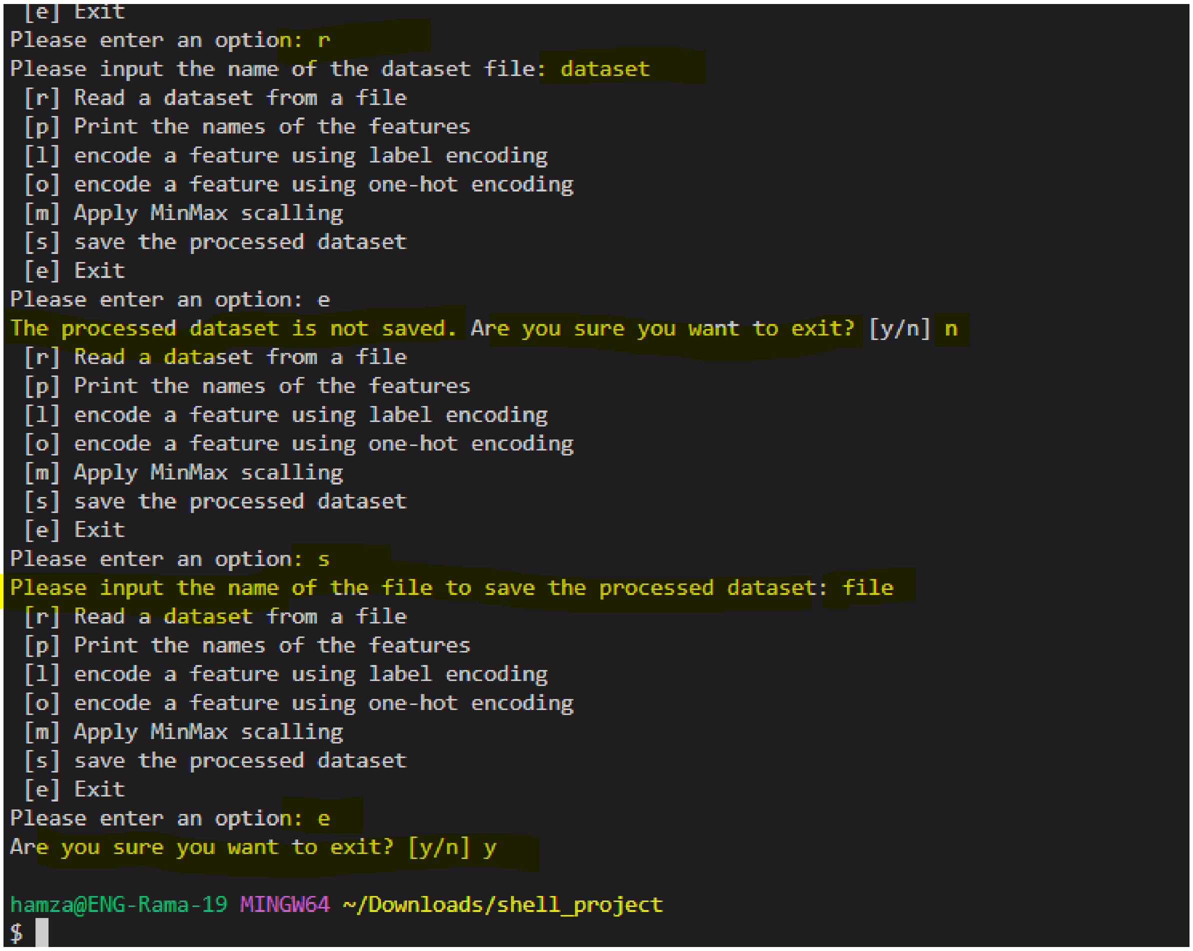
8. **If the user enters 'e':**

a. **The program should check if the processed dataset is saved using option “s”. if not, the program should print on the screen “The processed dataset is not saved. Are you sure you want to exist”. If the person inputs “yes”, the program ends. Otherwise, the program should return to main menu.**



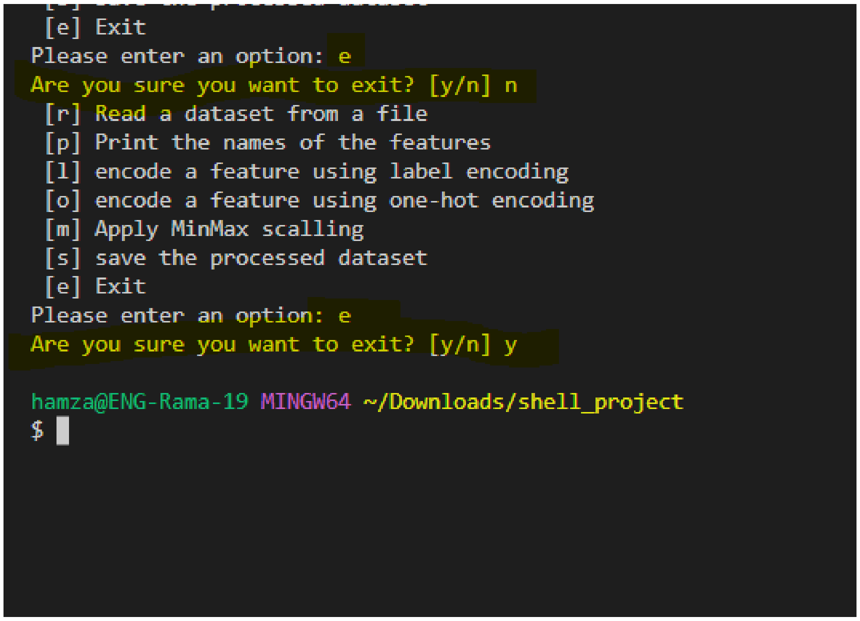
```
 [e] Exit
Please enter an option: e
The processed dataset is not saved. Are you sure you want to exit? [y/n] n
 [r] Read a dataset from a file
 [p] Print the names of the features
 [l] encode a feature using label encoding
 [o] encode a feature using one-hot encoding
 [m] Apply MinMax scalling
 [s] save the processed dataset
 [e] Exit
Please enter an option: e
The processed dataset is not saved. Are you sure you want to exit? [y/n] y

hamza@ENG-Rama-19 MINGW64 ~/Downloads/shell_project
```

After we returned to main menu we can choose to save file , then exit:



```
 [e] Exit
Please enter an option: r
Please input the name of the dataset file: dataset
 [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 scalling
 [s] save the processed dataset
 [e] Exit
Please enter an option: e
The processed dataset is not saved. Are you sure you want to exit? [y/n] n
 [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 scalling
 [s] save the processed dataset
 [e] Exit
Please enter an option: s
Please input the name of the file to save the processed dataset: file
 [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 scalling
 [s] save the processed dataset
 [e] Exit
Please enter an option: e
Are you sure you want to exit? [y/n] y

hamza@ENG-Rama-19 MINGW64 ~/Downloads/shell_project
$
```

b. **However, if the dataset is saved, the program should print on the screen "Are you sure you want to exist". If the person inputs "yes", the program ends. Otherwise, the program should return to the main menu.**



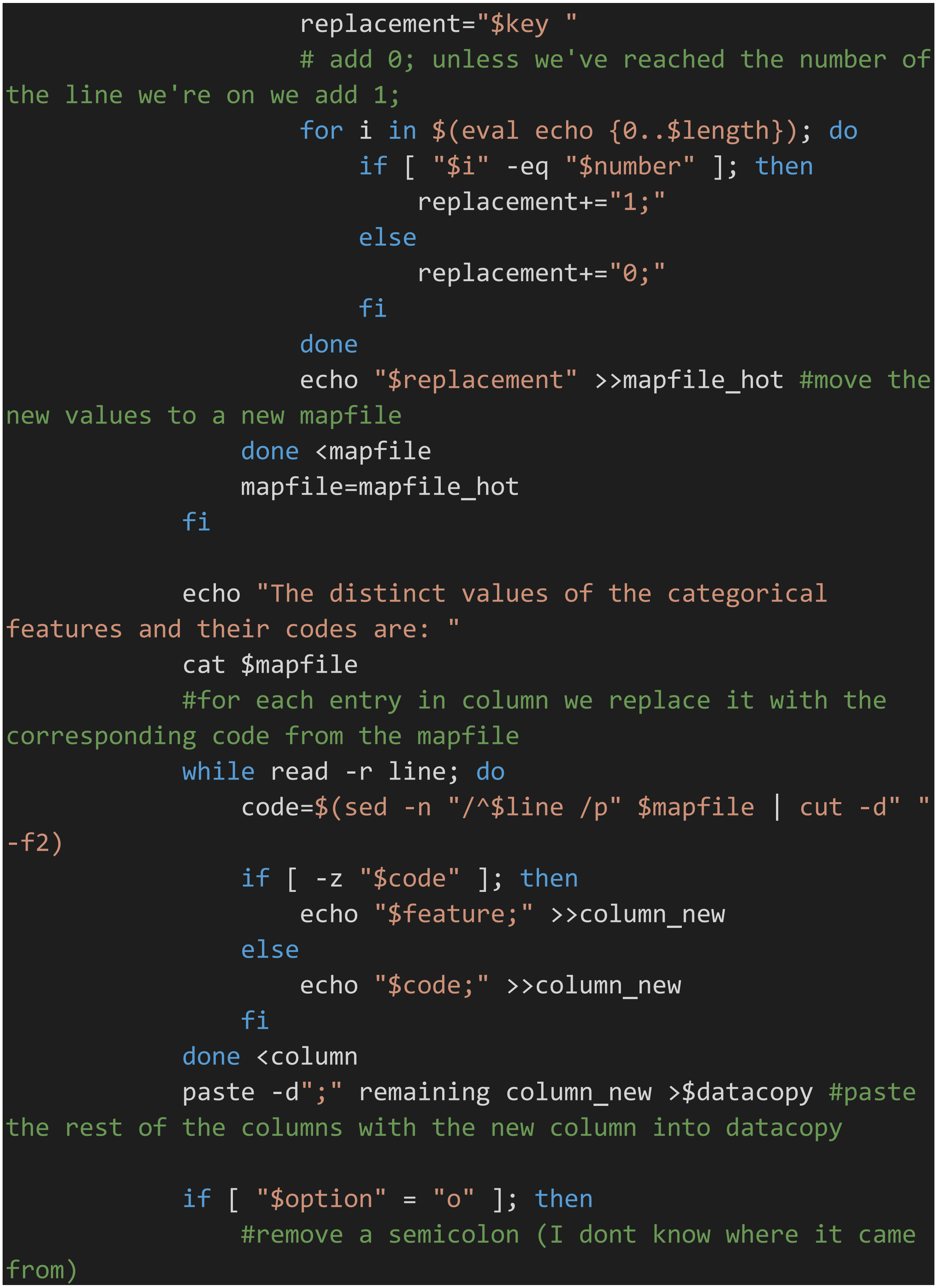
```
 [s] save the processed dataset
 [e] Exit
Please enter an option: e
Are you sure you want to exit? [y/n] n
 [r] Read a dataset from a file
 [p] Print the names of the features
 [l] encode a feature using label encoding
 [o] encode a feature using one-hot encoding
 [m] Apply MinMax scalling
 [s] save the processed dataset
 [e] Exit
Please enter an option: e
Are you sure you want to exit? [y/n] y

hamza@ENG-Rama-19 MINGW64 ~/Downloads/shell_project
$ 
```

Appendix :

The full code:

```
#!/bin/bash

function read_file {
    verified=0 #set verification flag to 0 since we are dealing
with a new file now
    if [ -e $datacopy ]; then
        rm $datacopy # delete the copy file
    fi
    read -p "Please input the name of the dataset file: "
dataset #read the name of the dataset file
    if [ ! -e "$dataset" ];
then                                  #if the file doesnt exist
        echo "The file doesn't exist"
        # check if the first and second line have the same
number of values
    elif [ "$(head -1 $dataset | sed "s/;/ /g" | wc -w)" !=
"$(head -2 $dataset | tail -1 | sed "s/;/ /g" | wc -w)" ]; then
        echo "The format of the data in the dataset file is
wrong"
    else #if everything is good, we set verified to 1 and copy
it to datacopy file
        cat $dataset >$datacopy
        verified=1
        saved=0 #set saved to 0 since its a new file now
    fi
}

function print_features {
    if [ "$verified" != 1 ]; then #check if file is verified
        echo "You must first read a dataset from a file"
    else # if its verified we print the features
        features=$(head -1 $datacopy)
        echo "The features are: $features"
    fi
}

function encoding {
    if [ "$verified" != 1 ]; then
        echo "You must first read a dataset from a file"
    else
        read -p "Please input the name of the categorical
feature for encoding: " feature
        # check if the feature name entered is in the first
line
        if [ "$(head -1 $datacopy | grep "$feature;")" !=
"$(head -1 $datacopy)" ]; then
            echo "The name of categorical feature is wrong"
        else
            index=$(head -1 $datacopy | tr ";" "\n" | grep -nx
$feature | cut -d":" -f1) #index of feature
            index_prev=$((index -
1))                                                      #index
of the previous feature
            index_next=$((index +
1))                                                      #index
of the next feature

            cut -d ';' -f$index $datacopy >column #the column
with the feature we're working on
            #the command to find the remaining features is
different for the first feature and the other features
            if [ "$index" == 1 ]; then
                cut -d ';' -f2- $datacopy >remaining
            else
                cut -d ';' -f1-$index_prev,$index_next-
$datacopy >remaining
            fi
            #remove the last semicolon from the end of each
line
```

```
            while read -r line; do
                line2=$(echo "$line" | sed 's/.$//')
                echo "$line2" >>remaining2
            done <remaining
            cat remaining2 >remaining
            rm remaining2

            tail -n +2 column | sort -u >unique       #remove
the first line (feature name) from column and remove duplicates
            awk '{print $0 " " NR-1}' unique >mapfile #create a
mapfile containing "entry line_number"
            mapfile=mapfile
            rm unique
            # one-hot encoding
            if [ "$option" = "o" ]; then
                # now we will take the first column from the
mapfile with the unique entries and insert feature_ to its
beginning
                #we also move it to a file called col
                sed -e "s/^\([^ ]*\)/$feature\_\1/" mapfile |
cut -d ' ' -f1 >>column_hot
                feature=$(cat column_hot | tr '\n' ';') #we
copy the column of these entries and create a line of them
seperated by semicolons since theyre our new features
                length=$(($(wc -l <mapfile) -
1))       #calculate the number of lines in the mapfile

                # for each line in map file, we replace the
number of the line with ones are zeros accordingly
                while read -r line; do
                    key=$(echo "$line" | awk '{print
$1}')    #value from the first column of the mapfile (name of
the entry)
                    number=$(echo "$line" | awk '{print $2}')
#number of the line (entry code using label encoding)
```



```
                    replacement="$key "
                    # add 0; unless we've reached the number of the line we're on we add 1;
                    for i in $(eval echo {0..$length}); do
                        if [ "$i" -eq "$number" ]; then
                            replacement+="1;"
                        else
                            replacement+="0;"
                        fi
                    done
                    echo "$replacement" >>mapfile_hot #move the new values to a new mapfile
                done <mapfile
                mapfile=mapfile_hot
            fi

            echo "The distinct values of the categorical features and their codes are: "
            cat $mapfile
            #for each entry in column we replace it with the corresponding code from the mapfile
            while read -r line; do
                code=$(sed -n "/^$line /p" $mapfile | cut -d" " -f2)
                if [ -z "$code" ]; then
                    echo "$feature;" >>column_new
                else
                    echo "$code;" >>column_new
                fi
            done <column
            paste -d";" remaining column_new >$datacopy #paste the rest of the columns with the new column into datacopy

            if [ "$option" = "o" ]; then
                #remove a semicolon (I dont know where it came from)
                while read -r line; do
                    line2=$(echo "$line" | sed 's/.$//')
                    echo "$line2" >>temp
                done <$datacopy
                cat temp >datacopy
                rm temp

                rm mapfile_hot
                rm column_hot
            fi
            #remove the temporary files created to run this
code
            rm remaining
            rm mapfile
            rm column_new
            rm column
        fi
    fi
}

function scaling {
    if [ "$verified" != 1 ]; then
        echo "You must first read a dataset from a file"
    else
        read -p "Please input the name of the feature to be
scaled: " feature
        if [ "$(head -1 $datacopy | grep "$feature;")" !=
"$(head -1 $datacopy)" ]; then
            echo "The name of categorical feature is wrong"
        else
            index=$(head -1 $datacopy | tr ";" "\n" | grep -nx
$feature | cut -d":" -f1)
            # num_check is the first entry of the feature, its
used to check if its categorical or not
            num_check=$(sed -n '2p' $datacopy | cut -d ';' -f
$index)
```

```
            if expr "$num_check" + 1 >/dev/null 2>&1; then #if
its a number, we scale
                index=$(head -1 $datacopy | tr ";" "\n" | grep
-nx $feature | cut -d":" -f1)
                index_prev=$((index - 1))
                index_next=$((index + 1))
                cut -d ';' -f$index $datacopy >column
                if [ "$index" == 1 ]; then
                    cut -d ';' -f2- $datacopy >remaining
                else
                    cut -d ';' -f1-$index_prev,$index_next-
$datacopy >remaining
                fi

                while read -r line; do
                    line2=$(echo "$line" | sed 's/.$//')
                    echo "$line2" >>remaining2
                done <remaining
                cat remaining2 >remaining
                rm remaining2
                min=$(tail +2 column | sort -n | head -
1)    #the minimum value from the column
                max=$(tail +2 column | sort -n -r | head -1)
#the maxiumum value from the column
                echo "Max: $max"
                echo "Min: $min"
                while read -r line; do
                    if [ "$line" == $feature ]; then
                        echo "$feature;" >>column_new
                    else
                        code=$(awk "BEGIN {print ($line - $min)
/ ($max - $min)}") #the equation to find the scaled value
                        echo "$code;"
>>column_new                                 #put new scaled
value into a new column
                    fi
                done <column
                paste -d";" remaining column_new >$datacopy
#paste the new data into datacopy

                rm remaining
                rm column_new
                rm column
            else #if its not a number, we print an error
mesasge
                echo "this feature is categorical feature and
must be encoded first"
            fi
        fi
    fi
}

datacopy="datacopy" # the name of the file where we store a
copy of the dataset
verified=0          # flag to indicate if a file has been
verified already or not
saved=0             # flag to indicate if a file has been saved
already or not

#program main menu is a loop
while true; do
    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 scalling "
    echo " [s] save the processed dataset"
    echo " [e] Exit"

    # read the user's input
    read -p "Please enter an option: " option
    # Reading the dataset file
    if [ "$option" = "r" ]; then
        read_file
        # Print features
    elif [ "$option" = "p" ]; then
        print_features
    # encoding
    elif [ "$option" == "l" ] || [ "$option" == "o" ]; then
        encoding
        # Scaling a feature
    elif [ "$option" = "m" ]; then
        scaling
        # Saving modified dataset to file
    elif [ "$option" = "s" ]; then
        if [ "$verified" != 1 ]; then
            echo "You must first read a dataset from a file"
        else
            read -p "Please input the name of the file to save the processed dataset: " savefile
            cat $datacopy >$savefile
            saved=1
        fi
        # exit
    elif [ "$option" = "e" ]; then
        if [ "$saved" = 1 ]; then
            read -p "Are you sure you want to exit? [y/n] " yesno
            if [ "$yesno" = "y" ]; then
                if [ -e $datacopy ]; then
                    rm $datacopy # delete the copy file
                fi
                exit 1
            fi
        else
            read -p "The processed dataset is not saved. Are you sure you want to exit? [y/n] " yesno
            if [ "$yesno" = "y" ]; then
                if [ -e $datacopy ]; then
                    rm $datacopy # delete the copy file
                fi
                exit 1
            fi
        fi
    else
        echo "Please select one of the options (r, p, l, o, m,
s, e)"
    fi
done
```