ADS-PHASE 4

ASSESSEMENT OF MARGINAL WORKERS IN TAMILNADU-A SOCIOECONIMC ANALYSIS

1. IMPORTING THE REQUIRED LIBRARIES
2. IMPORTING THE DATASET
3. CREATING A MATRIX
4. HANDALING THE MISSING DATAS
5. ENCODING THE CATEGORICAL DATA
6. SPLITTING THE DATA SET
7. FEATURE SCALING

CODING

[3]:
```
#importing the required libraries
import pandas as pd
import numpy as np
#to load the dataset
a=pd.read_csv("C:\\Users\\00AD␣
 ↪LAB\\Downloads\\DDW_B06ST_3300_State_TAMIL_NADU-2011.csv")
print(a)
```

```
    Table Code State Code District Code            Area Name   \
0      B0906ST        `33          `000   State - TAMIL NADU
1      B0906ST        `33          `000   State - TAMIL NADU
2      B0906ST        `33          `000   State - TAMIL NADU
3      B0906ST        `33          `000   State - TAMIL NADU
4      B0906ST        `33          `000   State - TAMIL NADU
..         ...        ...           ...                  ...
589    B0906ST        `33          `633  District - Tiruppur
590    B0906ST        `33          `633  District - Tiruppur
591    B0906ST        `33          `633  District - Tiruppur
592    B0906ST        `33          `633  District - Tiruppur
593    B0906ST        `33          `633  District - Tiruppur

    Total/ Rural/ Urban       Age group  \
0                 Total           Total
1                 Total            `5-14
2                 Total          15-34
3                 Total          35-59
```

```
4        Total        60+
..         ...        ...
589      Urban       `5-14
590      Urban      15-34
591      Urban      35-59
592      Urban        60+
593      Urban  Age not stated

   Worked for 3 months or more but less than 6 months -  Persons  \
0                                               66695
1                                                2637
2                                               31370
```

```
3                                                27418
4                                                 5219
..                                                 ...
589                                                  4
590                                                 54
591                                                 38
592                                                  4
593                                                  0

     Worked for 3 months or more but less than 6 months - Males  \
0                                                32578
1                                                 1345
2                                                15374
3                                                12976
4                                                 2865
..                                                 ...
589                                                  4
590                                                 35
591                                                 24
592                                                  2
593                                                  0

     Worked for 3 months or more but less than 6 months - Females  \
0                                                34117
1                                                 1292
2                                                15996
3                                                14442
4                                                 2354
..                                                 ...
589                                                  0
590                                                 19
591                                                 14
592                                                  2
593                                                  0

     Worked for less than 3 months - Persons  ...  \
0                                       12153  ...
1                                         356  ...
2                                        5714  ...
3                                        4757  ...
4                                        1320  ...
..                                        ...  ...
589                                         0  ...
590                                        14  ...
591                                         2  ...
592                                         0  ...
593                                         0  ...
```

```
     Industrial Category - N to O - Females  \
0                                       110
1                                         0
2                                        46
3                                        52
4                                        12
..                                      ...
589                                       0
590                                       0
591                                       0
592                                       0
593                                       0

     Industrial Category - P to Q - Persons  \
0                                       278
1                                        14
2                                       198
3                                        60
4                                         6
..                                      ...
589                                       0
590                                       4
591                                       2
592                                       0
593                                       0

     Industrial Category - P to Q - Males  \
0                                     128
1                                       6
2                                      94
3                                      24
4                                       4
..                                    ...
589                                     0
590                                     2
591                                     2
592                                     0
593                                     0

     Industrial Category - P to Q - Females\
0                                       150
1                                         8
2                                       104
3                                        36
4                                         2
..                                      ...
589                                       0
590                                       2
```

```
591                                        0
592                                        0
593                                        0

     Industrial Category - R to U - HHI - Persons\
0                                             978
1                                              36
2                                             508
3                                             356
4                                              78
..                                            ...
589                                             0
590                                             0
591                                             0
592                                             0
593                                             0

     Industrial Category - R to U - HHI - Males  \
0                                         226
1                                          16
2                                         114
3                                          68
4                                          28
..                                        ...
589                                         0
590                                         0
591                                         0
592                                         0
593                                         0

     Industrial Category - R to U - HHI - Females  \
0                                             752
1                                              20
2                                             394
3                                             288
4                                              50
..                                            ...
589                                             0
590                                             0
591                                             0
592                                             0
593                                             0

     Industrial Category - R to U - Non HHI - Persons  \
0                                                5152
1                                                1156
2                                                2602
3                                                1058
```

```
4                                                    318
..                                                   ...
589                                                    4
590                                                   12
591                                                   28
592                                                    4
593                                                    0

     Industrial Category - R to U - Non HHI - Males\
0                                                2270
1                                                 586
2                                                1178
3                                                 348
4                                                 152
..                                                ...
589                                                 4
590                                                 6
591                                                16
592                                                 2
593                                                 0

     Industrial Category - R to U - Non HHI - Females
0                                                  2882
1                                                   570
2                                                  1424
3                                                   710
4                                                   166
..                                                  ...
589                                                   0
590                                                   6
591                                                  12
592                                                   2
593                                                   0

[594 rows x 69 columns]
```

[18]:

```
import pandas as pd
import numpy as np
#to convert the dataset into array
a=pd.read_csv("C:\\Users\\OOAD_
  ↪LAB\\Downloads\\DDW_B06ST_3300_State_TAMIL_NADU-2011.csv")
array=a.to_numpy()
array1=np.array(a)
print(array)
print(array1)
```

```
[['B0906ST' '`33' '`000' ... 5152 2270 2882]
 ['B0906ST' '`33' '`000' ... 1156 586 570]
 ['B0906ST' '`33' '`000' … 2602 1178 1424]
 …
 ['B0906ST' '`33' '`633' … 28 16 12]
 ['B0906ST' '`33' '`633' … 4 2 2]
 ['B0906ST' '`33' '`633' … 0 0 0]]
[['B0906ST' '`33' '`000' … 5152 2270 2882]
 ['B0906ST' '`33' '`000' … 1156 586 570]
 ['B0906ST' '`33' '`000' … 2602 1178 1424]
 …
 ['B0906ST' '`33' '`633' … 28 16 12]
 ['B0906ST' '`33' '`633' … 4 2 2]
 ['B0906ST' '`33' '`633' … 0 0 0]]
```

[48]:

```
#to create the matric from the dataset:
a=pd.read_csv("C:\\Users\\OOAD␣
  ↪LAB\\Downloads\\DDW_B06ST_3300_State_TAMIL_NADU-2011.csv")
matrix = np.array([["Worked for less than 3 months - Persons"], ["Industrial␣
  ↪Category - P to Q - Persons"]])
print(matrix[0, 0])
#print(matrix[1, 1])
print("Shape of the matrix:", matrix.shape)  # Output: (3, 3)

# Matrix transpose
transpose_matrix = array1.T
print("Transposed matrix:")
print(transpose_matrix)
```

```
Worked for less than 3 months - Persons
Shape of the matrix: (2, 1)
Transposed matrix:
[['B0906ST' 'B0906ST' 'B0906ST' ... 'B0906ST' 'B0906ST' 'B0906ST']
 ['`33' '`33' '`33' ... '`33' '`33' '`33']
 ['`000' '`000' '`000' ... '`633' '`633' '`633']
 …
 [5152 1156 2602 ... 28 4 0]
 [2270 586 1178 ... 16 2 0]
 [2882 570 1424 ... 12 2 0]]
```

[11]:

```
from sklearn.impute import SimpleImputer
# to handle the dataset with missing values
data = np.array([[1, 2, np.nan], [4, np.nan, 6], [7, 8, 9]])

# Create an instance of the SimpleImputer class
imputer = SimpleImputer(strategy='mean')  # Other strategies: 'median',␣
  ↪'most_frequent', 'constant'

# Fit the imputer to the data and transform it
```

```
imputed_data = imputer.fit_transform(data)

# The missing values have been replaced
print("Original Data:")
print(data)

print("\nImputed Data:")
print(imputed_data)
```

```
Original Data:
[[ 1.  2. nan]
 [ 4. nan  6.]
 [ 7.  8.  9.]]

Imputed Data:
[[1.  2.  7.5]
 [4.  5.  6. ]
 [7.  8.  9. ]]
```

[38]:
```
#encoding the categorical dataset:
from sklearn.preprocessing import OneHotEncoder
import numpy as np
import pandas as pd

a=pd.read_csv("C:\\Users\\OOAD␣
  ↪LAB\\Downloads\\DDW_B06ST_3300_State_TAMIL_NADU-2011.csv")


# Sample data
data = ['Age group','Worked for less than 3 months - Persons','Industrial␣
  ↪Category - P to Q - Females']

# Initialize the OneHotEncoder
onehot_encoder = OneHotEncoder(sparse=False)  # Use sparse=False to get a dense␣
  ↪matrix

# Fit and transform the data (reshape is necessary)
encoded_data = onehot_encoder.fit_transform(np.array(data).reshape(-1, 1))

print("Original data:", data)
print("One-Hot Encoded data:")
print(encoded_data)
```

```
Original data: ['Age group', 'Worked for less than 3 months - Persons',
'Industrial Category - P to Q - Females']
One-Hot Encoded data:
[[1. 0. 0.]
```

```
 [0. 0. 1.]
 [0. 1. 0.]]
```

[37]:

```
#feature scalling process:
from sklearn.preprocessing import StandardScaler
import numpy as np
a=pd.read_csv("C:\\Users\\OOAD␣
↪LAB\\Downloads\\DDW_B06ST_3300_State_TAMIL_NADU-2011.csv")
# Sample data
data= np.array(mydataset["Worked for less than 3 months - Persons"]).
↪reshape(-1,1)

# Initialize the StandardScaler
scaler = StandardScaler()

# Fit and transform the data
scaled_data = scaler.fit_transform(data)

# Print the scaled data
print("Original data:")
print(data)
print("Scaled data:")
print(scaled_data)
```

```
Original data:
[[12153]
 [  356]
 [ 5714]
 [ 4757]
 [ 1320]
 [    6]
 [10306]
 [  334]
 [ 4817]
 [ 3995]
 [ 1154]
 [    6]
 [ 1847]
 [   22]
 [  897]
 [  762]
 [  166]
 [    0]
 [ 1366]
 [   41]
 [  636]
 [  556]
 [  131]
[    2]
[ 1226]
[   39]
[  571]
[  497]
[  117]
[    2]
[  140]
[    2]
[   65]
[   59]
[   14]
[    0]
[   66]
[    2]
[   37]
[   27]
[    0]
[    0]
[    0]
[    0]
[    0]
[    0]
[    0]
[    0]
[   66]
[    2]
[   37]
[   27]
[    0]
[    0]
[  998]
[   14]
[  477]
[  413]
[   94]
[    0]
[  764]
[   13]
[  354]
[  324]
[   73]
[    0]
[  234]
[    1]
[  123]
[   89]
[   21]
[    0]
[ 2346]
[   69]
[ 1146]
[  926]
[  203]
[    2]
[ 2210]
[   67]
[ 1082]
[  876]
[  183]
[    2]
[  136]
[    2]
[   64]
[   50]
[   20]
[    0]
[ 1635]
[   57]
[  787]
[  597]
[  194]
[    0]
[ 1605]
[   53]
[  773]
[  587]
[  192]
[    0]
[   30]
[    4]
[   14]
[   10]
[    2]
[    0]
[ 1186]
[   27]
[  573]
[  455]
[  130]
[    1]
[ 1111]
[   27]
[  538]
[  427]
[  118]
[     1]
[    75]
[     0]
[    35]
[    28]
[    12]
[     0]
[   534]
[    17]
[   233]
[   181]
[   103]
[     0]
[   515]
[    17]
[   225]
[   175]
[    98]
[     0]
[    19]
[     0]
[     8]
[     6]
[     5]
[     0]
[   172]
[    15]
[    78]
[    43]
[    36]
[     0]
[   172]
[    15]
[    78]
[    43]
[    36]
[     0]
[     0]
[     0]
[     0]
[     0]
[     0]
[     0]
[   205]
[     5]
[    83]
[    85]
[    32]
[     0]
[   203]
[     3]
[    83]
[    85]
[    32]
[     0]
[     2]
[     2]
[     0]
[     0]
[     0]
[     0]
[   551]
[    11]
[   291]
[   202]
[    46]
[     1]
[   312]
[    11]
[   148]
[   117]
[    35]
[     1]
[   239]
[     0]
[   143]
[    85]
[    11]
[     0]
[   114]
[     3]
[    53]
[    39]
[    19]
[     0]
[   110]
[     3]
[    51]
[    37]
[    19]
[     0]
[     4]
[     0]
[     2]
[     2]
[     0]
[      0]
[      2]
[      0]
[      0]
[      0]
[      2]
[      0]
[      0]
[      0]
[      0]
[      0]
[      0]
[      0]
[      2]
[      0]
[      0]
[      0]
[      2]
[      0]
[     44]
[      0]
[     24]
[     10]
[     10]
[      0]
[     28]
[      0]
[     12]
[      6]
[     10]
[      0]
[     16]
[      0]
[     12]
[      4]
[      0]
[      0]
[     77]
[      1]
[     43]
[     31]
[      2]
[      0]
[     22]
[      1]
[     17]
[      3]
[      1]
[      0]
[     55]
[      0]
[     26]
[     28]
[      1]
[      0]
[    211]
[      7]
[     76]
[     91]
[     37]
[      0]
[    207]
[      7]
[     74]
[     91]
[     35]
[      0]
[      4]
[      0]
[      2]
[      0]
[      2]
[      0]
[    457]
[     13]
[    213]
[    205]
[     26]
[      0]
[    259]
[     13]
[    119]
[    110]
[     17]
[      0]
[    198]
[      0]
[     94]
[     95]
[      9]
[      0]
[     63]
[      3]
[     20]
[     36]
[      4]
[      0]
[     24]
[      2]
[      6]
[     13]
[      3]
[      0]
[     39]
[      1]
[     14]
[     23]
[      1]
[      0]
[     24]
[      0]
[     10]
[     13]
[      1]
[      0]
[     17]
[      0]
[      5]
[     11]
[      1]
[      0]
[      7]
[      0]
[      5]
[      2]
[      0]
[      0]
[     18]
[      0]
[      7]
[      8]
[      3]
[      0]
[     14]
[      0]
[      3]
[      8]
[      3]
[      0]
[      4]
[      0]
[      4]
[      0]
[      0]
[     0]
[    51]
[     6]
[    26]
[    17]
[     2]
[     0]
[    51]
[     6]
[    26]
[    17]
[     2]
[     0]
[     0]
[     0]
[     0]
[     0]
[     0]
[     0]
[    21]
[     1]
[     2]
[     9]
[     9]
[     0]
[     3]
[     1]
[     0]
[     1]
[     1]
[     0]
[    18]
[     0]
[     2]
[     8]
[     8]
[     0]
[    66]
[     0]
[    26]
[    31]
[     9]
[     0]
[    32]
[     0]
[     8]
[    15]
[     9]
[     0]
[    34]
[     0]
[    18]
[    16]
[     0]
[     0]
[     2]
[     0]
[     2]
[     0]
[     0]
[     0]
[     2]
[     0]
[     2]
[     0]
[     0]
[     0]
[     0]
[     0]
[     0]
[     0]
[     0]
[     0]
[    11]
[     0]
[     6]
[     3]
[     2]
[     0]
[     9]
[     0]
[     4]
[     3]
[     2]
[     0]
[     2]
[     0]
[     2]
[     0]
[     0]
[     0]
[    13]
[     0]
[     6]
[     7]
[     0]
[    0]
[    8]
[    0]
[    4]
[    4]
[    0]
[    0]
[    5]
[    0]
[    2]
[    3]
[    0]
[    0]
[   12]
[    2]
[    3]
[    7]
[    0]
[    0]
[    9]
[    2]
[    2]
[    5]
[    0]
[    0]
[    3]
[    0]
[    1]
[    2]
[    0]
[    0]
[  179]
[    3]
[   69]
[   65]
[   42]
[    0]
[   30]
[    1]
[   12]
[    8]
[    9]
[    0]
[  149]
[    2]
[   57]
[   57]
[   33]
[      0]
[    274]
[      4]
[    113]
[    132]
[     25]
[      0]
[     31]
[      0]
[      8]
[     11]
[     12]
[      0]
[    243]
[      4]
[    105]
[    121]
[     13]
[      0]
[    574]
[     30]
[    270]
[    198]
[     76]
[      0]
[    556]
[     30]
[    262]
[    190]
[     74]
[      0]
[     18]
[      0]
[      8]
[      8]
[      2]
[      0]
[    313]
[     14]
[    144]
[    120]
[     35]
[      0]
[    298]
[     14]
[    139]
[    112]
[     33]
```

```
 [   0]
 [  15]
 [   0]
 [   5]
 [   8]
 [   2]
 [   0]
 [ 551]
 [  11]
 [ 245]
 [ 248]
 [  47]
 [   0]
 [ 477]
 [   9]
 [ 210]
 [ 219]
 [  39]
 [   0]
 [  74]
 [   2]
 [  35]
 [  29]
 [   8]
 [   0]
 [  17]
 [   0]
 [  15]
 [   2]
 [   0]
 [   0]
 [   1]
 [   0]
 [   1]
 [   0]
 [   0]
 [   0]
 [  16]
 [   0]
 [  14]
 [   2]
 [   0]
 [   0]]
Scaled data:
[[ 1.50388006e+01]
 [ 2.41240535e-01]
 [ 6.96204472e+00]
 [ 5.76163233e+00]
[ 1.45043337e+00]
[-1.97781762e-01]
[ 1.27220173e+01]
[ 2.13644848e-01]
[ 5.83689329e+00]
[ 4.80581807e+00]
[ 1.24221137e+00]
[-1.97781762e-01]
[ 2.11147552e+00]
[-1.77712171e-01]
[ 9.19843570e-01]
[ 7.50506399e-01]
[ 2.91414512e-03]
[-2.05307858e-01]
[ 1.50813345e+00]
[-1.53879532e-01]
[ 5.92458372e-01]
[ 4.92110419e-01]
[-4.09880846e-02]
[-2.02799160e-01]
[ 1.33252453e+00]
[-1.56388231e-01]
[ 5.10925660e-01]
[ 4.18103803e-01]
[-5.85489764e-02]
[-2.02799160e-01]
[-2.96989398e-02]
[-2.02799160e-01]
[-1.23775146e-01]
[-1.31301243e-01]
[-1.87746967e-01]
[-2.05307858e-01]
[-1.22520797e-01]
[-2.02799160e-01]
[-1.58896930e-01]
[-1.71440424e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-1.22520797e-01]
[-2.02799160e-01]
[-1.58896930e-01]
[-1.71440424e-01]
```

```
[-2.05307858e-01]
[-2.05307858e-01]
[ 1.04653286e+00]
[-1.87746967e-01]
[  3.93016814e-01]
[  3.12738452e-01]
[-8.73990131e-02]
[-2.05307858e-01]
[  7.53015098e-01]
[-1.89001316e-01]
[  2.38731836e-01]
[  2.01101353e-01]
[-1.13740351e-01]
[-2.05307858e-01]
[  8.82099056e-02]
[-2.04053509e-01]
[-5.10228799e-02]
[-9.36707602e-02]
[-1.78966521e-01]
[-2.05307858e-01]
[ 2.73739588e+00]
[-1.18757749e-01]
[ 1.23217658e+00]
[  9.56219704e-01]
[  4.93250736e-02]
[-2.02799160e-01]
[ 2.56680436e+00]
[-1.21266447e-01]
[ 1.15189821e+00]
[  8.93502233e-01]
[  2.42380852e-02]
[-2.02799160e-01]
[-3.47163375e-02]
[-2.02799160e-01]
[-1.25029496e-01]
[-1.42590387e-01]
[-1.80220870e-01]
[-2.05307858e-01]
[ 1.84555344e+00]
[-1.33809942e-01]
[  7.81865134e-01]
[  5.43538745e-01]
[  3.80359288e-02]
[-2.05307858e-01]
[ 1.80792296e+00]
[-1.38827339e-01]
[  7.64304243e-01]
[ 5.30995251e-01]
[ 3.55272300e-02]
[-2.05307858e-01]
[-1.67677376e-01]
[-2.00290461e-01]
[-1.87746967e-01]
[-1.92764364e-01]
[-2.02799160e-01]
[-2.05307858e-01]
[ 1.28235055e+00]
[-1.71440424e-01]
[ 5.13434359e-01]
[ 3.65421127e-01]
[-4.22424340e-02]
[-2.04053509e-01]
[ 1.18827435e+00]
[-1.71440424e-01]
[ 4.69532129e-01]
[ 3.30299343e-01]
[-5.72946270e-02]
[-2.04053509e-01]
[-1.11231652e-01]
[-2.05307858e-01]
[-1.61405629e-01]
[-1.70186075e-01]
[-1.90255665e-01]
[-2.05307858e-01]
[ 4.64514731e-01]
[-1.83983918e-01]
[ 8.69555562e-02]
[ 2.17293864e-02]
[-7.61098683e-02]
[-2.05307858e-01]
[ 4.40682092e-01]
[-1.83983918e-01]
[ 7.69207608e-02]
[ 1.42032899e-02]
[-8.23816154e-02]
[-2.05307858e-01]
[-1.81475219e-01]
[-2.05307858e-01]
[-1.95273063e-01]
[-1.97781762e-01]
[-1.99036111e-01]
[-2.05307858e-01]
[ 1.04402416e-02]
[-1.86492617e-01]
[-1.07468604e-01]
[-1.51370833e-01]
```

```
[-1.60151279e-01]
[-2.05307858e-01]
[ 1.04402416e-02]
[-1.86492617e-01]
[-1.07468604e-01]
[-1.51370833e-01]
[-1.60151279e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[ 5.18337725e-02]
[-1.99036111e-01]
[-1.01196857e-01]
[-9.86881578e-02]
[-1.65168677e-01]
[-2.05307858e-01]
[ 4.93250736e-02]
[-2.01544810e-01]
[-1.01196857e-01]
[-9.86881578e-02]
[-1.65168677e-01]
[-2.05307858e-01]
[-2.02799160e-01]
[-2.02799160e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[ 4.85838671e-01]
[-1.91510015e-01]
[ 1.59707823e-01]
[ 4.80707242e-02]
[-1.47607785e-01]
[-2.04053509e-01]
[ 1.86049160e-01]
[-1.91510015e-01]
[-1.96641444e-02]
[-5.85489764e-02]
[-1.61405629e-01]
[-2.04053509e-01]
[ 9.44816527e-02]
[-2.05307858e-01]
[-2.59358915e-02]
[-9.86881578e-02]
```

```
[-1.91510015e-01]
[-2.05307858e-01]
[-6.23120247e-02]
[-2.01544810e-01]
[-1.38827339e-01]
[-1.56388231e-01]
[-1.81475219e-01]
[-2.05307858e-01]
[-6.73294224e-02]
[-2.01544810e-01]
[-1.41336038e-01]
[-1.58896930e-01]
[-1.81475219e-01]
[-2.05307858e-01]
[-2.00290461e-01]
[-2.05307858e-01]
[-2.02799160e-01]
[-2.02799160e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-2.02799160e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-2.02799160e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-2.02799160e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-2.02799160e-01]
[-2.05307858e-01]
[-1.50116484e-01]
[-2.05307858e-01]
[-1.75203472e-01]
[-1.92764364e-01]
[-1.92764364e-01]
[-2.05307858e-01]
[-1.70186075e-01]
[-2.05307858e-01]
[-1.90255665e-01]
[-1.97781762e-01]
```

```
[-1.92764364e-01]
[-2.05307858e-01]
[-1.85238268e-01]
[-2.05307858e-01]
[-1.90255665e-01]
[-2.00290461e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-1.08722953e-01]
[-2.04053509e-01]
[-1.51370833e-01]
[-1.66423026e-01]
[-2.02799160e-01]
[-2.05307858e-01]
[-1.77712171e-01]
[-2.04053509e-01]
[-1.83983918e-01]
[-2.01544810e-01]
[-2.04053509e-01]
[-2.05307858e-01]
[-1.36318640e-01]
[-2.05307858e-01]
[-1.72694774e-01]
[-1.70186075e-01]
[-2.04053509e-01]
[-2.05307858e-01]
[  5.93598690e-02]
[-1.96527413e-01]
[-1.09977303e-01]
[-9.11620613e-02]
[-1.58896930e-01]
[-2.05307858e-01]
[  5.43424713e-02]
[-1.96527413e-01]
[-1.12486001e-01]
[-9.11620613e-02]
[-1.61405629e-01]
[-2.05307858e-01]
[-2.00290461e-01]
[-2.05307858e-01]
[-2.02799160e-01]
[-2.05307858e-01]
[-2.02799160e-01]
[-2.05307858e-01]
[  3.67929826e-01]
[-1.89001316e-01]
[  6.18685678e-02]
[ 5.18337725e-02]
```

```
[-1.72694774e-01]
[-2.05307858e-01]
[ 1.19568641e-01]
[-1.89001316e-01]
[-5.60402776e-02]
[-6.73294224e-02]
[-1.83983918e-01]
[-2.05307858e-01]
[ 4.30533265e-02]
[-2.05307858e-01]
[-8.73990131e-02]
[-8.61446636e-02]
[-1.94018714e-01]
[-2.05307858e-01]
[-1.26283845e-01]
[-2.01544810e-01]
[-1.80220870e-01]
[-1.60151279e-01]
[-2.00290461e-01]
[-2.05307858e-01]
[-1.75203472e-01]
[-2.02799160e-01]
[-1.97781762e-01]
[-1.89001316e-01]
[-2.01544810e-01]
[-2.05307858e-01]
[-1.56388231e-01]
[-2.04053509e-01]
[-1.87746967e-01]
[-1.76457822e-01]
[-2.04053509e-01]
[-2.05307858e-01]
[-1.75203472e-01]
[-2.05307858e-01]
[-1.92764364e-01]
[-1.89001316e-01]
[-2.04053509e-01]
[-2.05307858e-01]
[-1.83983918e-01]
[-2.05307858e-01]
[-1.99036111e-01]
[-1.91510015e-01]
[-2.04053509e-01]
[-2.05307858e-01]
[-1.96527413e-01]
[-2.05307858e-01]
[-1.99036111e-01]
[-2.02799160e-01]
```

```
[-2.05307858e-01]
[-2.05307858e-01]
[-1.82729569e-01]
[-2.05307858e-01]
[-1.96527413e-01]
[-1.95273063e-01]
[-2.01544810e-01]
[-2.05307858e-01]
[-1.87746967e-01]
[-2.05307858e-01]
[-2.01544810e-01]
[-1.95273063e-01]
[-2.01544810e-01]
[-2.05307858e-01]
[-2.00290461e-01]
[-2.05307858e-01]
[-2.00290461e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-1.41336038e-01]
[-1.97781762e-01]
[-1.72694774e-01]
[-1.83983918e-01]
[-2.02799160e-01]
[-2.05307858e-01]
[-1.41336038e-01]
[-1.97781762e-01]
[-1.72694774e-01]
[-1.83983918e-01]
[-2.02799160e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-1.78966521e-01]
[-2.04053509e-01]
[-2.02799160e-01]
[-1.94018714e-01]
[-1.94018714e-01]
[-2.05307858e-01]
[-2.01544810e-01]
[-2.04053509e-01]
[-2.05307858e-01]
[-2.04053509e-01]
```

```
[-2.04053509e-01]
[-2.05307858e-01]
[-1.82729569e-01]
[-2.05307858e-01]
[-2.02799160e-01]
[-1.95273063e-01]
[-1.95273063e-01]
[-2.05307858e-01]
[-1.22520797e-01]
[-2.05307858e-01]
[-1.72694774e-01]
[-1.66423026e-01]
[-1.94018714e-01]
[-2.05307858e-01]
[-1.65168677e-01]
[-2.05307858e-01]
[-1.95273063e-01]
[-1.86492617e-01]
[-1.94018714e-01]
[-2.05307858e-01]
[-1.62659978e-01]
[-2.05307858e-01]
[-1.82729569e-01]
[-1.85238268e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-2.02799160e-01]
[-2.05307858e-01]
[-2.02799160e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-2.02799160e-01]
[-2.05307858e-01]
[-2.02799160e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-1.91510015e-01]
[-2.05307858e-01]
[-1.97781762e-01]
[-2.01544810e-01]
```

```
[-2.02799160e-01]
[-2.05307858e-01]
[-1.94018714e-01]
[-2.05307858e-01]
[-2.00290461e-01]
[-2.01544810e-01]
[-2.02799160e-01]
[-2.05307858e-01]
[-2.02799160e-01]
[-2.05307858e-01]
[-2.02799160e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-1.89001316e-01]
[-2.05307858e-01]
[-1.97781762e-01]
[-1.96527413e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-1.95273063e-01]
[-2.05307858e-01]
[-2.00290461e-01]
[-2.00290461e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-1.99036111e-01]
[-2.05307858e-01]
[-2.02799160e-01]
[-2.01544810e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-1.90255665e-01]
[-2.02799160e-01]
[-2.01544810e-01]
[-1.96527413e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-1.94018714e-01]
[-2.02799160e-01]
[-2.02799160e-01]
[-1.99036111e-01]
[-2.05307858e-01]
[-2.05307858e-01]
[-2.01544810e-01]
[-2.05307858e-01]
[-2.04053509e-01]
[-2.02799160e-01]
```

```
[-2.05307858e-01]
[-2.05307858e-01]
[  1.92206876e-02]
[-2.01544810e-01]
[-1.18757749e-01]
[-1.23775146e-01]
[-1.52625183e-01]
[-2.05307858e-01]
[-1.67677376e-01]
[-2.04053509e-01]
[-1.90255665e-01]
[-1.95273063e-01]
[-1.94018714e-01]
[-2.05307858e-01]
[-1.84097950e-02]
[-2.02799160e-01]
[-1.33809942e-01]
[-1.33809942e-01]
[-1.63914328e-01]
[-2.05307858e-01]
[  1.38383882e-01]
[-2.00290461e-01]
[-6.35663741e-02]
[-3.97337351e-02]
[-1.73949123e-01]
[-2.05307858e-01]
[-1.66423026e-01]
[-2.05307858e-01]
[-1.95273063e-01]
[-1.91510015e-01]
[-1.90255665e-01]
[-2.05307858e-01]
[  9.94990504e-02]
[-2.00290461e-01]
[-7.36011694e-02]
[-5.35315787e-02]
[-1.89001316e-01]
[-2.05307858e-01]
[  5.14688708e-01]
[-1.67677376e-01]
[  1.33366485e-01]
[  4.30533265e-02]
[-1.09977303e-01]
[-2.05307858e-01]
[  4.92110419e-01]
[-1.67677376e-01]
[  1.23331689e-01]
[ 3.30185312e-02]
```

```
[-1.12486001e-01]
[-2.05307858e-01]
[-1.82729569e-01]
[-2.05307858e-01]
[-1.95273063e-01]
[-1.95273063e-01]
[-2.02799160e-01]
[-2.05307858e-01]
[ 1.87303510e-01]
[-1.87746967e-01]
[-2.46815421e-02]
[-5.47859282e-02]
[-1.61405629e-01]
[-2.05307858e-01]
[ 1.68488268e-01]
[-1.87746967e-01]
[-3.09532892e-02]
[-6.48207235e-02]
[-1.63914328e-01]
[-2.05307858e-01]
[-1.86492617e-01]
[-2.05307858e-01]
[-1.99036111e-01]
[-1.95273063e-01]
[-2.02799160e-01]
[-2.05307858e-01]
[ 4.85838671e-01]
[-1.91510015e-01]
[ 1.02007749e-01]
[ 1.05770797e-01]
[-1.46353436e-01]
[-2.05307858e-01]
[ 3.93016814e-01]
[-1.94018714e-01]
[ 5.81055196e-02]
[ 6.93946643e-02]
[-1.56388231e-01]
[-2.05307858e-01]
[-1.12486001e-01]
[-2.02799160e-01]
[-1.61405629e-01]
[-1.68931725e-01]
[-1.95273063e-01]
[-2.05307858e-01]
[-1.83983918e-01]
[-2.05307858e-01]
[-1.86492617e-01]
[-2.02799160e-01]
```

```
 [-2.05307858e-01]
 [-2.05307858e-01]
 [-2.04053509e-01]
 [-2.05307858e-01]
 [-2.04053509e-01]
 [-2.05307858e-01]
 [-2.05307858e-01]
 [-2.05307858e-01]
 [-1.85238268e-01]
 [-2.05307858e-01]
 [-1.87746967e-01]
 [-2.02799160e-01]
 [-2.05307858e-01]
 [-2.05307858e-01]]
```

[21]:

```
#to create the input array for train & test process: x=np.array(mydataset["Worked
for less than 3 months - Males"]).reshape(-1,1) print(x)
```

```
[[5414]
 [ 184]
 [2511]
 [2072]
 [ 647]
 [   0]
 [4536]
 [ 173]
 [2085]
 [1734]
 [ 544]
 [   0]
 [ 878]
 [  11]
 [ 426]
 [ 338]
 [ 103]
 [   0]
 [ 647]
 [  21]
 [ 284]
 [ 263]
 [  79]
 [   0]
 [ 565]
 [  19]
 [ 247]
 [ 229]
 [  70]
[   0]
[  82]
[   2]
[  37]
[  34]
[   9]
[   0]
[  38]
[   0]
[  22]
[  16]
[   0]
[   0]
[   0]
[   0]
[   0]
[   0]
[   0]
[   0]
[  38]
[   0]
[  22]
[  16]
[   0]
[   0]
[ 507]
[  10]
[ 229]
[ 223]
[  45]
[   0]
[ 374]
[   9]
[ 162]
[ 169]
[  34]
[   0]
[ 133]
[   1]
[  67]
[  54]
[  11]
[   0]
[1110]
[  28]
[ 543]
[ 433]
[ 106]
[   0]
[1032]
[  26]
[ 507]
[ 408]
[  91]
[   0]
[  78]
[   2]
[  36]
[  25]
[  15]
[   0]
[ 652]
[  28]
[ 314]
[ 240]
[  70]
[   0]
[ 644]
[  28]
[ 313]
[ 235]
[  68]
[   0]
[   8]
[   0]
[   1]
[   5]
[   2]
[   0]
[ 470]
[  19]
[ 209]
[ 175]
[  67]
[   0]
[ 446]
[  19]
[ 199]
[ 167]
[  61]
[   0]
[  24]
[   0]
[  10]
[   8]
[   6]
[   0]
[ 241]
[  12]
[ 103]
[  77]
[  49]
[   0]
[ 231]
[  12]
[  97]
[  77]
[  45]
[   0]
[  10]
[   0]
[   6]
[   0]
[   4]
[   0]
[  61]
[   6]
[  30]
[  10]
[  15]
[   0]
[  61]
[   6]
[  30]
[  10]
[  15]
[   0]
[   0]
[   0]
[   0]
[   0]
[   0]
[   0]
[  89]
[   3]
[  38]
[  34]
[  14]
[   0]
[  87]
[   1]
[  38]
[  34]
[  14]
[   0]
[   2]
[   2]
[   0]
[   0]
[   0]
[   0]
[ 261]
[   8]
[ 143]
[  88]
[  22]
[   0]
[ 137]
[   8]
[  70]
[  43]
[  16]
[   0]
[ 124]
[   0]
[  73]
[  45]
[   6]
[   0]
[  54]
[   2]
[  25]
[  16]
[  11]
[   0]
[  54]
[   2]
[  25]
[  16]
[  11]
[   0]
[   0]
[   0]
[   0]
[   0]
[   0]
[   0]
[   0]
[   0]
[   0]
[   0]
[   0]
[     0]
[     0]
[     0]
[     0]
[     0]
[     0]
[     0]
[     0]
[     0]
[     0]
[     0]
[     0]
[     0]
[    20]
[     0]
[    12]
[     4]
[     4]
[     0]
[    10]
[     0]
[     4]
[     2]
[     4]
[     0]
[    10]
[     0]
[     8]
[     2]
[     0]
[     0]
[    35]
[     0]
[    19]
[    15]
[     1]
[     0]
[     8]
[     0]
[     7]
[     1]
[     0]
[     0]
[    27]
[     0]
[    12]
[    14]
[     1]
[   0]
[ 100]
[   7]
[  33]
[  42]
[  18]
[   0]
[  98]
[   7]
[  32]
[  42]
[  17]
[   0]
[   2]
[   0]
[   1]
[   0]
[   1]
[   0]
[ 152]
[   3]
[  71]
[  62]
[  16]
[   0]
[ 119]
[   3]
[  52]
[  51]
[  13]
[   0]
[  33]
[   0]
[  19]
[  11]
[   3]
[   0]
[  30]
[   3]
[  12]
[  13]
[   2]
[   0]
[  10]
[   2]
[   2]
[   5]
[   1]
[   0]
[  20]
[   1]
[  10]
[   8]
[   1]
[   0]
[  19]
[   0]
[   8]
[  10]
[   1]
[   0]
[  13]
[   0]
[   4]
[   8]
[   1]
[   0]
[   6]
[   0]
[   4]
[   2]
[   0]
[   0]
[   8]
[   0]
[   2]
[   3]
[   3]
[   0]
[   6]
[   0]
[   0]
[   3]
[   3]
[   0]
[   2]
[   0]
[   2]
[   0]
[   0]
[   0]
[  40]
[   6]
[  18]
[  14]
[   2]
[    0]
[   40]
[    6]
[   18]
[   14]
[    2]
[    0]
[    0]
[    0]
[    0]
[    0]
[    0]
[    0]
[   10]
[    0]
[    0]
[    2]
[    8]
[    0]
[    0]
[    0]
[    0]
[    0]
[    0]
[    0]
[   10]
[    0]
[    0]
[    2]
[    8]
[    0]
[   19]
[    0]
[    4]
[   12]
[    3]
[    0]
[   13]
[    0]
[    2]
[    8]
[    3]
[    0]
[    6]
[    0]
[    2]
[    4]
[    0]
[     0]
[     2]
[     0]
[     2]
[     0]
[     0]
[     0]
[     2]
[     0]
[     2]
[     0]
[     0]
[     0]
[     0]
[     0]
[     0]
[     0]
[     0]
[     0]
[     6]
[     0]
[     4]
[     1]
[     1]
[     0]
[     4]
[     0]
[     2]
[     1]
[     1]
[     0]
[     2]
[     0]
[     2]
[     0]
[     0]
[     0]
[     6]
[     0]
[     2]
[     4]
[     0]
[     0]
[     4]
[     0]
[     2]
[     2]
[     0]
[    0]
[    2]
[    0]
[    0]
[    2]
[    0]
[    0]
[    5]
[    0]
[    0]
[    5]
[    0]
[    0]
[    5]
[    0]
[    0]
[    5]
[    0]
[    0]
[    0]
[    0]
[    0]
[    0]
[    0]
[    0]
[  107]
[    0]
[   50]
[   32]
[   25]
[    0]
[   19]
[    0]
[    9]
[    6]
[    4]
[    0]
[   88]
[    0]
[   41]
[   26]
[   21]
[    0]
[  139]
[    2]
[   54]
[   68]
[   15]
[   0]
[  11]
[   0]
[   2]
[   0]
[   9]
[   0]
[ 128]
[   2]
[  52]
[  68]
[   6]
[   0]
[ 187]
[  14]
[  97]
[  52]
[  24]
[   0]
[ 180]
[  14]
[  94]
[  48]
[  24]
[   0]
[   7]
[   0]
[   3]
[   4]
[   0]
[   0]
[ 161]
[   7]
[  71]
[  60]
[  23]
[   0]
[ 155]
[   7]
[  68]
[  59]
[  21]
[   0]
[   6]
[   0]
[   3]
[   1]
[   2]
[    0]
[  237]
[    5]
[  111]
[   98]
[   23]
[    0]
[  207]
[    4]
[   96]
[   91]
[   16]
[    0]
[   30]
[    1]
[   15]
[    7]
[    7]
[    0]
[    1]
[    0]
[    1]
[    0]
[    0]
[    0]
[    1]
[    0]
[    1]
[    0]
[    0]
[    0]
[    0]
[    0]
[    0]
[    0]
[    0]
[    0]]
```

[22]:
```
x.shape
```

[22]:
```
(594, 1)
```

[23]:
```
#to create the output array for train & test process:
y=np.array(mydataset["Worked for less than 3 months - Persons"])
print(y)
```

```
[12153   356  5714  4757  1320     6 10306   334  4817  3995  1154     6
  1847    22   897   762   166     0  1366    41   636   556   131     2
  1226    39   571   497   117     2   140     2    65    59    14     0
    66     2    37    27     0     0     0     0     0     0     0     0
    66     2    37    27     0     0   998    14   477   413    94     0
   764    13   354   324    73     0   234     1   123    89    21     0
  2346    69  1146   926   203     2  2210    67  1082   876   183     2
   136     2    64    50    20     0  1635    57   787   597   194     0
  1605    53   773   587   192     0    30     4    14    10     2     0
  1186    27   573   455   130     1  1111    27   538   427   118     1
    75     0    35    28    12     0   534    17   233   181   103     0
   515    17   225   175    98     0    19     0     8     6     5     0
   172    15    78    43    36     0   172    15    78    43    36     0
     0     0     0     0     0     0   205     5    83    85    32     0
   203     3    83    85    32     0     2     2     0     0     0     0
   551    11   291   202    46     1   312    11   148   117    35     1
   239     0   143    85    11     0   114     3    53    39    19     0
   110     3    51    37    19     0     4     0     2     2     0     0
     2     0     0     0     2     0     0     0     0     0     0     0
     2     0     0     0     2     0    44     0    24    10    10     0
    28     0    12     6    10     0    16     0    12     4     0     0
    77     1    43    31     2     0    22     1    17     3     1     0
    55     0    26    28     1     0   211     7    76    91    37     0
   207     7    74    91    35     0     4     0     2     0     2     0
   457    13   213   205    26     0   259    13   119   110    17     0
   198     0    94    95     9     0    63     3    20    36     4     0
    24     2     6    13     3     0    39     1    14    23     1     0
    24     0    10    13     1     0    17     0     5    11     1     0
     7     0     5     2     0     0    18     0     7     8     3     0
    14     0     3     8     3     0     4     0     4     0     0     0
    51     6    26    17     2     0    51     6    26    17     2     0
     0     0     0     0     0     0    21     1     2     9     9     0
     3     1     0     1     1     0    18     0     2     8     8     0
    66     0    26    31     9     0    32     0     8    15     9     0
    34     0    18    16     0     0     2     0     2     0     0     0
     2     0     2     0     0     0     0     0     0     0     0     0
    11     0     6     3     2     0     9     0     4     3     2     0
     2     0     2     0     0     0    13     0     6     7     0     0
     8     0     4     4     0     0     5     0     2     3     0     0
    12     2     3     7     0     0     9     2     2     5     0     0
     3     0     1     2     0     0   179     3    69    65    42     0
    30     1    12     8     9     0   149     2    57    57    33     0
   274     4   113   132    25     0    31     0     8    11    12     0
   243     4   105   121    13     0   574    30   270   198    76     0
   556    30   262   190    74     0    18     0     8     8     2     0
   313    14   144   120    35     0   298    14   139   112    33     0
    15     0     5     8     2     0   551    11   245   248    47     0
   477     9   210   219    39     0    74     2    35    29     8     0
    17     0    15     2     0     0     1     0     1     0     0     0
    16     0    14     2     0     0]
```

[24]:

```
y.shape
```

[24]:

```
(594,)
```

[25]:

```
print(type(x))
```

```
<class 'numpy.ndarray'>
```

[26]:

```
print(type(y))
```

```
<class 'numpy.ndarray'>
```

[29]:

```
#splitting the dataset into test set & training set:
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20)
x_train.shape
```

[29]:

```
(475, 1)
```

[30]:

```
x_test.shape
```

[30]:

```
(119, 1)
```

[31]:

```
y.shape
```

[31]:

```
(594,)
```

[32]:

```
print(y_train.shape)
```

```
(475,)
```

[33]:

```
print(y_test.shape)
```

```
(119,)
```

[35]:

```
#spliting & training dataset with mean squared error:
# Initialize the model
model = LinearRegression()

# Train the model
model.fit(x_train, y_train)

# Make predictions on the test set
y_pred = model.predict(x_test)

# Evaluate the model
```

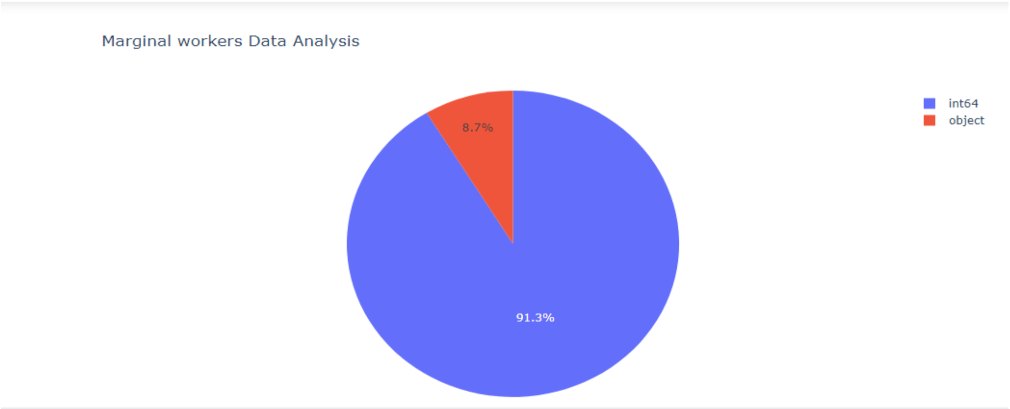
```
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print("Mean Squared Error:", mse)
print("R-squared:", r2)
```

```
Mean Squared Error: 796.7282889466414
R-squared: 0.9919039963931663
```

[49]:

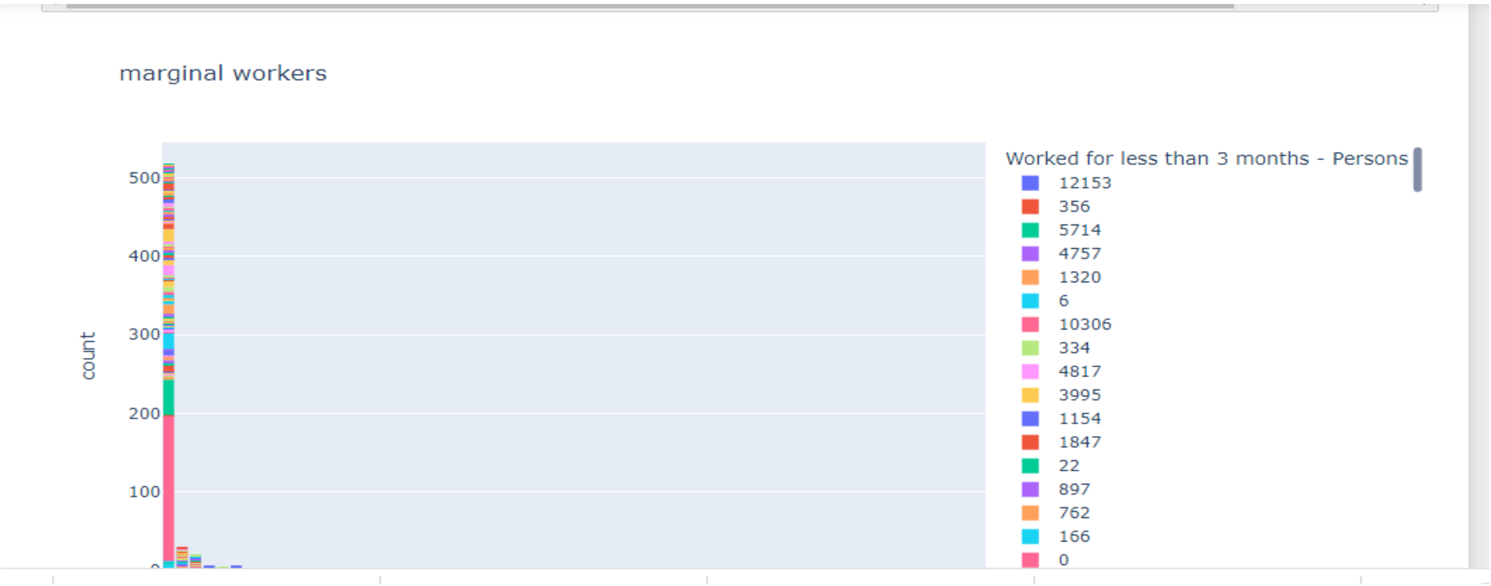
```
#pictorical representation for the dataset by using  pie chart:
import plotly.graph_objects as go
data_types_counts = mydataset.dtypes.astype(str).value_counts()
fig = go.Figure(data=[go.Pie(labels=data_types_counts.index,␣
 ↪values=data_types_counts.values)])
fig.update_layout(title_text=" Marginal workers Data Analysis")
fig.show()
```

Output



[43]:

```
#pictorical representation for the dataset by using  bar chart:
import plotly.express as px
fig = px.histogram(mydataset,x="Worked for less than 3 months - Persons",␣
 ↪title="marginal workers", color="Worked for less than 3 months - Persons")
# Update the layout and add box plots
fig.update_layout
    (bargap=0.2)

fig.show()
```


marginal workers
count
500
400
300
200
100
0
Worked for less than 3 months - Persons
12153
356
5714
4757
1320
6
10306
334
4817
3995
1154
1847
22
897
762
166
0