| | |
|---|---|
| Semester | T.E. Semester V – Computer Engineering |
| Subject | Data Warehousing and Mining |
| Subject Professor In-charge | Dr. Kavita P Shirsat |
| Laboratory | M-312A |
| Student Name | Musab Khan |
| Roll Number | 22102A0066 |
| Experiment Number | 01 |
| Experiment Title | Pre-processing of Dataset to remove missing values using central tendency (Mean, median mode) |
| Resources / Apparatus Required | Hardware: Computer system |
| Description | • The terms mean, median and mode are used to describe the central tendency of a large data set. Range provides<br>• When working with a large data set, it can be useful to represent the entire data set with a single value that descr mode are all ways to describe it.<br>• To find the mean, add up the values in the data set and then divide by the number of values that you added.<br>• To find the median, list the values of the data set in numerical order and identify which value appears in the middl<br>• To find the mode, identify which value in the data set occurs most often.<br>• Missing values are identified and replaced by the mean median or mode based on the requirement. |
| Link of the | import pandas as pd<br>import numpy as np<br>data = pd.read_csv('adult.csv')<br>data = data.applymap(lambda x: np.nan if str(x).strip() == '?' else x)<br>print(data.head(10))<br>missing_values = data.isnull().sum()<br>print(missing_values)<br>numerical_cols = [<br>'age', 'fnlwgt', 'education-num', 'capital-gain', 'capital-loss',<br>'hours-per-week'<br>]<br>for col in numerical_cols:<br>data[col].fillna(data[col].mean(), inplace=True)<br>categorical_cols = [<br>'workclass', 'education', 'marital-status', 'occupation', 'relationship',<br>'race', 'sex', 'native-country', 'income'<br>] |

```
for col in categorical_cols:

data[col].fillna(data[col].mode()[0], inplace=True)

missing_values = data.isnull().sum()

print(missing_values)

new_filename = 'cleaned_dataset.csv'

data.to_csv(new_filename, index=False)

print(f"Cleaned data saved to {new_filename}")
```

```
/home/runner/DWM/main.py:6: FutureWarning: DataFrame.applymap has been
  data = data.applymap(lambda x: np.nan if str(x).strip() == '?' else
   age          workclass  fnlwgt   education  education-num  ... capi
0   39          State-gov   77516   Bachelors             13  ...
1   50   Self-emp-not-inc   83311   Bachelors             13  ...
2   38                NaN  215646     HS-grad              9  ...
3   53            Private  234721        11th              7  ...
4   28            Private  338409   Bachelors             13  ...
5   37            Private  284582     Masters             14  ...
6   49            Private  160187         9th              5  ...
7   52   Self-emp-not-inc  209642     HS-grad              9  ...
8   31            Private   45781     Masters             14  ...
9   42            Private  159449   Bachelors             13  ...

[10 rows x 15 columns]
age                0
workclass       1837
fnlwgt             0
education          0
education-num      0
marital-status     0
occupation      1843
relationship       0
race               0
sex                0
capital-gain       0
capital-loss       0
hours-per-week     0
native-country   583
income             0
dtype: int64
/home/runner/DWM/main.py:17: FutureWarning: A value is trying to be se
 method.
The behavior will change in pandas 3.0. This inplace method will never
ves as a copy.
```

Output

Run

```
ves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try usi
stead, to perform the operation inplace on the original object.


  data[col].fillna(data[col].mean(), inplace=True)
/home/runner/DWM/main.py:25: FutureWarning: A value is trying to be se
 method.
The behavior will change in pandas 3.0. This inplace method will never
ves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try usi
stead, to perform the operation inplace on the original object.


  data[col].fillna(data[col].mode()[0], inplace=True)
age               0
workclass         0
fnlwgt            0
education         0
education-num     0
marital-status    0
occupation        0
relationship      0
race              0
sex               0
capital-gain      0
capital-loss      0
hours-per-week    0
native-country    0
income            0
dtype: int64
Cleaned data saved to cleaned_dataset.csv
```