ML - Lab Assingment 3
MLP Classifier

[1]:
```
import pandas as pd
import numpy as np
```

[63]:
```
df=pd.read_csv(r'D:\ML Lab\adult (1).csv')
```

[3]:
```
#info
#describe
#ead tail cor
```

[4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48842 entries, 0 to 48841
Data columns (total 15 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   age              48842 non-null  int64
 1   workclass        48842 non-null  object
 2   fnlwgt           48842 non-null  int64
 3   education        48842 non-null  object
 4   educational-num  48842 non-null  int64
 5   marital-status   48842 non-null  object
 6   occupation       48842 non-null  object
 7   relationship     48842 non-null  object
 8   race             48842 non-null  object
 9   gender           48842 non-null  object
 10  capital-gain     48842 non-null  int64
 11  capital-loss     48842 non-null  int64
 12  hours-per-week   48842 non-null  int64
 13  native-country   48842 non-null  object
 14  income           48842 non-null  object
dtypes: int64(6), object(9)
memory usage: 5.6+ MB
```

[5]:
```
df.head()
```

[5]:

```
   age  workclass  fnlwgt     education  educational-num      marital-status  \
0   25    Private  226802          11th                7       Never-married
1   38    Private   89814       HS-grad                9  Married-civ-spouse
2   28  Local-gov  336951    Assoc-acdm               12  Married-civ-spouse
3   44    Private  160323  Some-college               10  Married-civ-spouse
4   18          ?  103497  Some-college               10       Never-married

          occupation relationship   race  gender  capital-gain  capital-loss  \
0  Machine-op-inspct    Own-child  Black    Male             0             0
1    Farming-fishing      Husband  White    Male             0             0
2    Protective-serv      Husband  White    Male             0             0
3  Machine-op-inspct      Husband  Black    Male          7688             0
4                  ?    Own-child  White  Female             0             0

   hours-per-week native-country income
0              40  United-States  <=50K
1              50  United-States  <=50K
2              40  United-States   >50K
3              40  United-States   >50K
4              30  United-States  <=50K
```

[6]:

```
df.tail(10)
```

[6]:

```
       age     workclass  fnlwgt     education  educational-num  \
48832   32       Private   34066          10th                6
48833   43       Private   84661     Assoc-voc               11
48834   32       Private  116138       Masters               14
48835   53       Private  321865       Masters               14
48836   22       Private  310152  Some-college               10
48837   27       Private  257302    Assoc-acdm               12
48838   40       Private  154374       HS-grad                9
48839   58       Private  151910       HS-grad                9
48840   22       Private  201490       HS-grad                9
48841   52  Self-emp-inc  287927       HS-grad                9

           marital-status         occupation   relationship  \
48832  Married-civ-spouse  Handlers-cleaners        Husband
48833  Married-civ-spouse              Sales        Husband
48834       Never-married       Tech-support  Not-in-family
48835  Married-civ-spouse    Exec-managerial        Husband
48836       Never-married    Protective-serv  Not-in-family
48837  Married-civ-spouse       Tech-support           Wife
48838  Married-civ-spouse  Machine-op-inspct        Husband
48839             Widowed       Adm-clerical      Unmarried
48840       Never-married       Adm-clerical      Own-child
48841  Married-civ-spouse    Exec-managerial           Wife
```

```
                    race  gender  capital-gain  capital-loss  hours-per-week  \
48832  Amer-Indian-Eskimo    Male             0             0              40
48833               White    Male             0             0              45
48834  Asian-Pac-Islander    Male             0             0              11
48835               White    Male             0             0              40
48836               White    Male             0             0              40
48837               White  Female             0             0              38
48838               White    Male             0             0              40
48839               White  Female             0             0              40
48840               White    Male             0             0              20
48841               White  Female         15024             0              40

      native-country income
48832  United-States  <=50K
48833  United-States  <=50K
48834         Taiwan  <=50K
48835  United-States   >50K
48836  United-States  <=50K
48837  United-States  <=50K
48838  United-States   >50K
48839  United-States  <=50K
48840  United-States  <=50K
48841  United-States   >50K
```

[7]:
```
df.tail(10)
```

[7]:
```
       age     workclass  fnlwgt     education  educational-num  \
48832   32       Private   34066          10th                6
48833   43       Private   84661     Assoc-voc               11
48834   32       Private  116138       Masters               14
48835   53       Private  321865       Masters               14
48836   22       Private  310152  Some-college               10
48837   27       Private  257302    Assoc-acdm               12
48838   40       Private  154374       HS-grad                9
48839   58       Private  151910       HS-grad                9
48840   22       Private  201490       HS-grad                9
48841   52  Self-emp-inc  287927       HS-grad                9

           marital-status         occupation   relationship  \
48832  Married-civ-spouse  Handlers-cleaners        Husband
48833  Married-civ-spouse              Sales        Husband
48834       Never-married       Tech-support  Not-in-family
48835  Married-civ-spouse    Exec-managerial        Husband
48836       Never-married    Protective-serv  Not-in-family
48837  Married-civ-spouse       Tech-support           Wife
48838  Married-civ-spouse  Machine-op-inspct        Husband
48839             Widowed       Adm-clerical      Unmarried
```

```
48840       Never-married       Adm-clerical      Own-child
48841  Married-civ-spouse    Exec-managerial           Wife

                      race  gender  capital-gain  capital-loss  hours-per-week  \
48832   Amer-Indian-Eskimo    Male             0             0              40
48833                White    Male             0             0              45
48834   Asian-Pac-Islander    Male             0             0              11
48835                White    Male             0             0              40
48836                White    Male             0             0              40
48837                White  Female             0             0              38
48838                White    Male             0             0              40
48839                White  Female             0             0              40
48840                White    Male             0             0              20
48841                White  Female         15024             0              40

      native-country income
48832  United-States  <=50K
48833  United-States  <=50K
48834         Taiwan  <=50K
48835  United-States   >50K
48836  United-States  <=50K
48837  United-States  <=50K
48838  United-States   >50K
48839  United-States  <=50K
48840  United-States  <=50K
48841  United-States   >50K
```

[8]:

```python
df.describe()
```

[8]:

```
                age        fnlwgt  educational-num  capital-gain  \
count  48842.000000  4.884200e+04     48842.000000  48842.000000
mean      38.643585  1.896641e+05        10.078089   1079.067626
std       13.710510  1.056040e+05         2.570973   7452.019058
min       17.000000  1.228500e+04         1.000000      0.000000
25%       28.000000  1.175505e+05         9.000000      0.000000
50%       37.000000  1.781445e+05        10.000000      0.000000
75%       48.000000  2.376420e+05        12.000000      0.000000
max       90.000000  1.490400e+06        16.000000  99999.000000

       capital-loss  hours-per-week
count  48842.000000    48842.000000
mean      87.502314       40.422382
std      403.004552       12.391444
min        0.000000        1.000000
25%        0.000000       40.000000
50%        0.000000       40.000000
75%        0.000000       45.000000
```

```
max      4356.000000       99.000000
```

[9]:
```
df.corr()
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[9], line 1
----> 1 df.corr()

File c:\Users\karpe\anaconda3\envs\ml_lab\lib\site-packages\pandas\core\frame.py:
↪10704, in DataFrame.corr(self, method, min_periods, numeric_only)
  10702 cols = data.columns
  10703 idx = cols.copy()
> 10704 mat = data.to_numpy(dtype=float, na_value=np.nan, copy=False)
  10706 if method == "pearson":
  10707     correl = libalgos.nancorr(mat, minp=min_periods)

File c:\Users\karpe\anaconda3\envs\ml_lab\lib\site-packages\pandas\core\frame.py:
↪1889, in DataFrame.to_numpy(self, dtype, copy, na_value)
   1887 if dtype is not None:
   1888     dtype = np.dtype(dtype)
-> 1889 result = self._mgr.as_array(dtype=dtype, copy=copy, na_value=na_value)
   1890 if result.dtype is not dtype:
   1891     result = np.array(result, dtype=dtype, copy=False)

File c:
↪\Users\karpe\anaconda3\envs\ml_lab\lib\site-packages\pandas\core\internals\managers.
↪py:1656, in BlockManager.as_array(self, dtype, copy, na_value)
   1654         arr.flags.writeable = False
   1655 else:
-> 1656     arr = self._interleave(dtype=dtype, na_value=na_value)
   1657     # The underlying data was copied within _interleave, so no need
   1658     # to further copy if copy=True or setting na_value
   1660 if na_value is lib.no_default:

File c:
↪\Users\karpe\anaconda3\envs\ml_lab\lib\site-packages\pandas\core\internals\managers.
↪py:1715, in BlockManager._interleave(self, dtype, na_value)
   1713     else:
   1714         arr = blk.get_values(dtype)
-> 1715     result[rl.indexer] = arr
   1716     itemmask[rl.indexer] = 1
   1718 if not itemmask.all():

ValueError: could not convert string to float: 'Private'
```

[ ]:
```
df.shape
```

[ ]: (48842, 15)

[64]:
```
df.isna() #finds out is there any null values a
```

[64]:
```
         age  workclass  fnlwgt  education  educational-num  marital-status  \
0      False      False   False      False            False           False
1      False      False   False      False            False           False
2      False      False   False      False            False           False
3      False      False   False      False            False           False
4      False      False   False      False            False           False
…        …          …       …          …                …               …
48837  False      False   False      False            False           False
48838  False      False   False      False            False           False
48839  False      False   False      False            False           False
48840  False      False   False      False            False           False
48841  False      False   False      False            False           False

       occupation  relationship   race  gender  capital-gain  capital-loss  \
0           False         False  False   False         False         False
1           False         False  False   False         False         False
2           False         False  False   False         False         False
3           False         False  False   False         False         False
4           False         False  False   False         False         False
…             …             …      …       …             …             …
48837       False         False  False   False         False         False
48838       False         False  False   False         False         False
48839       False         False  False   False         False         False
48840       False         False  False   False         False         False
48841       False         False  False   False         False         False

       hours-per-week  native-country  income
0               False           False   False
1               False           False   False
2               False           False   False
3               False           False   False
4               False           False   False
…                 …               …       …
48837           False           False   False
48838           False           False   False
48839           False           False   False
48840           False           False   False
48841           False           False   False

[48842 rows x 15 columns]
```

[65]:
```
df.isna().sum() #finds out the total summ of the nulll values
```

[65]:
```
age               0
workclass         0
fnlwgt            0
education         0
educational-num   0
marital-status    0
occupation        0
relationship      0
race              0
gender            0
capital-gain      0
capital-loss      0
hours-per-week    0
native-country    0
income            0
dtype: int64
```

[66]:
```python
df.duplicated().sum() #duplicates records are given and sum-> count
```

[66]:
```
52
```

[13]:
```python
df=df.drop_duplicates() #drops the duplicates
```

[14]:
```python
df.duplicated().sum()
```

[14]:
```
0
```

[69]:
```python
df.isin(['?']).sum() #gives the data which are having ?
```

[69]:
```
age               0
workclass         0
fnlwgt            0
education         0
educational-num   0
marital-status    0
occupation        0
relationship      0
race              0
gender            0
capital-gain      0
capital-loss      0
hours-per-week    0
native-country    0
income            0
dtype: int64
```

[68]:
```
#can't drop this data coz dropping is feasible only till 10 datasets
#handling the missing values
#1)leave as it is
#2) fill the missing values
#3) drop missing values
df=df.replace('?',np.nan)
```

[60]:
```
df=df.isin(['?']).sum()
```

[17]:
```
df.isna().sum()
```

[17]:
```
age                   0
workclass          2795
fnlwgt                0
education             0
educational-num       0
marital-status        0
occupation         2805
relationship          0
race                  0
gender                0
capital-gain          0
capital-loss          0
hours-per-week        0
native-country      856
income                0
dtype: int64
```

[19]:
```
df.isna().sum()
```

[19]:
```
0
```

[20]:
```
df
```

[20]:
```
age                0
workclass          0
fnlwgt             0
education          0
educational-num    0
marital-status     0
occupation         0
relationship       0
race               0
gender             0
capital-gain       0
capital-loss       0
hours-per-week     0
```

```
native-country     0
income             0
dtype: int64
```

[21]:
```
temp=pd.DataFrame({
    "Name":['Abc','PQR',np.nan],
    "Roll no":[1,np.nan,3]
})
```

[22]:
```
temp
```

[22]:

	Name	Roll no
0	Abc	1.0
1	PQR	NaN
2	NaN	3.0

[23]:
```
temp.dropna(axis=0,inplace=False) #dropping out the null values "Nan"
```

[23]:

	Name	Roll no
0	Abc	1.0

[24]:
```
temp.fillna(method='bfill',inplace=True) #to fill the values of the nul values
```

```
C:\Users\karpe\AppData\Local\Temp\ipykernel_20696\2159169994.py:1:
FutureWarning: DataFrame.fillna with 'method' is deprecated and will raise in a
future version. Use obj.ffill() or obj.bfill() instead.
  temp.fillna(method='bfill',inplace=True) #to fill the values of the nul values
```

[25]:
```
temp
```

[25]:

	Name	Roll no
0	Abc	1.0
1	PQR	3.0
2	NaN	3.0

[26]:
```
temp.fillna(method='ffill',inplace=True)
```

```
C:\Users\karpe\AppData\Local\Temp\ipykernel_20696\2967702086.py:1:
FutureWarning: DataFrame.fillna with 'method' is deprecated and will raise in a
future version. Use obj.ffill() or obj.bfill() instead.
  temp.fillna(method='ffill',inplace=True)
```

[27]:
```
temp
```

[27]:

	Name	Roll no
0	Abc	1.0
1	PQR	3.0

```
2  PQR      3.0
```

[70]:
```
from sklearn.impute import SimpleImputer
imputer= SimpleImputer(strategy='most_frequent',missing_values=np.nan)
```

[84]:
```
#categorical data-> most_frequent,  fit-transfrom-> finds all the null values␣
 ↪and also find out all  the frequent null values and replace it
df['workclass']=imputer.fit_transform(df[['workclass']]).ravel() #ravel->␣
 ↪converrts into 1d
df['occupation']=imputer.fit_transform(df[['occupation']]).ravel()
df['native-country']=imputer.fit_transform(df[['native-country']]).ravel()
```

[71]:
```
df.isna().sum()
```

[71]:
```
age                  0
workclass         2799
fnlwgt               0
education            0
educational-num      0
marital-status       0
occupation        2809
relationship         0
race                 0
gender               0
capital-gain         0
capital-loss         0
hours-per-week       0
native-country     857
income               0
dtype: int64
```

[72]:
```
df['gender'].unique()
```

[72]:
```
array(['Male', 'Female'], dtype=object)
```

[73]:
```
df['gender'] = df['gender'].replace('Male',1)
df['gender'] = df['gender'].replace('Female',0)
```

[41]:
```
temp_df=pd.DataFrame({
    'Fruit_name':['Mango','Apple','Grapes','Pears'],
    'Fruit_color':['Red','Yellow','Orange','Yellow'],
    'Fruit_price':[1000,300,20,300]
})
```

[42]:
```
temp_df
```

[42]:
```
  Fruit_name Fruit_color  Fruit_price
0      Mango         Red         1000
1      Apple      Yellow          300
2     Grapes      Orange           20
3      Pears      Yellow          300
```

[49]:
```
from sklearn.preprocessing import LabelEncoder #randomly assigns the number
lbl_encoder=LabelEncoder()
temp_df['Fruit_name']=lbl_encoder.fit_transform(temp_df["Fruit_name"])
```

[ ]:
```
temp_df=pd.get_dummies(temp_df,columns=['Fruit_color'])
```

[ ]:
```
temp_df
```

[ ]:
```
   Fruit_name  Fruit_price  Fruit_color_Orange  Fruit_color_Red  \
0           2         1000               False             True
1           0          300               False            False
2           1           20                True            False
3           3          300               False            False

   Fruit_color_Yellow
0               False
1                True
2               False
3                True
```

[74]:
```
print(df['income'].unique())
```

```
['<=50K' '>50K']
```

[75]:
```
df['income']=df['income'].replace('<=50K',0)
df['income']=df['income'].replace('>50K',1)
```

[76]:
```
df
```

[76]:
```
       age     workclass  fnlwgt     education  educational-num  \
0       25       Private  226802          11th                7
1       38       Private   89814       HS-grad                9
2       28     Local-gov  336951    Assoc-acdm               12
3       44       Private  160323  Some-college               10
4       18           NaN  103497  Some-college               10
…     …              …       …             …               …
48837   27       Private  257302    Assoc-acdm               12
48838   40       Private  154374       HS-grad                9
48839   58       Private  151910       HS-grad                9
48840   22       Private  201490       HS-grad                9
48841   52  Self-emp-inc  287927       HS-grad                9
```

```
          marital-status         occupation relationship   race  gender  \
0          Never-married  Machine-op-inspct    Own-child  Black       1
1     Married-civ-spouse    Farming-fishing      Husband  White       1
2     Married-civ-spouse    Protective-serv      Husband  White       1
3     Married-civ-spouse  Machine-op-inspct      Husband  Black       1
4          Never-married                NaN    Own-child  White       0
…                    …                  …          …      …     …
48837 Married-civ-spouse       Tech-support         Wife  White       0
48838 Married-civ-spouse  Machine-op-inspct      Husband  White       1
48839            Widowed       Adm-clerical    Unmarried  White       0
48840      Never-married       Adm-clerical    Own-child  White       1
48841 Married-civ-spouse    Exec-managerial         Wife  White       0

       capital-gain  capital-loss  hours-per-week native-country  income
0                 0             0              40  United-States       0
1                 0             0              50  United-States       0
2                 0             0              40  United-States       1
3              7688             0              40  United-States       1
4                 0             0              30  United-States       0
…               …             …             …            …     …
48837             0             0              38  United-States       0
48838             0             0              40  United-States       1
48839             0             0              40  United-States       0
48840             0             0              20  United-States       0
48841         15024             0              40  United-States       1

[48842 rows x 15 columns]
```

[45]:
```
df.head()
```

[45]:
```
   age  workclass  fnlwgt     education  educational-num      marital-status  \
0   25    Private  226802          11th                7       Never-married
1   38    Private   89814       HS-grad                9  Married-civ-spouse
2   28  Local-gov  336951    Assoc-acdm               12  Married-civ-spouse
3   44    Private  160323  Some-college               10  Married-civ-spouse
4   18          ?  103497  Some-college               10       Never-married

          occupation relationship   race  gender  capital-gain  capital-loss  \
0  Machine-op-inspct    Own-child  Black       1             0             0
1    Farming-fishing      Husband  White       1             0             0
2    Protective-serv      Husband  White       1             0             0
3  Machine-op-inspct      Husband  Black       1          7688             0
4                  ?    Own-child  White       0             0             0

   hours-per-week native-country  income
0              40  United-States       0
```

```
1              50  United-States       0
2              40  United-States       1
3              40  United-States       1
4              30  United-States       0
```

[77]:
```
print(df['marital-status'].unique())
```

```
['Never-married' 'Married-civ-spouse' 'Widowed' 'Divorced' 'Separated'
 'Married-spouse-absent' 'Married-AF-spouse']
```

[78]:
```
df['marital-status']=df['marital-status'].replace('Never-married','Unmarried')
df['marital-status']=df['marital-status'].replace('Married-AF-spouse','Married')
df['marital-status']=df['marital-status'].
 ↪replace('Married-civ-spouse','Married')
df['marital-status']=df['marital-status'].
 ↪replace('Married-spouse-absent','Married')
df['marital-status']=df['marital-status'].replace('Separated','Separated')
df['marital-status']=df['marital-status'].replace('Divorced','Separated')
df['marital-status']=df['marital-status'].replace('Widowed','Widowed')
```

[79]:
```
df['marital-status']=lbl_encoder.fit_transform(df['marital-status'])
```

[80]:
```
df
```

[80]:
```
       age     workclass  fnlwgt     education  educational-num  \
0       25       Private  226802          11th                7
1       38       Private   89814       HS-grad                9
2       28     Local-gov  336951    Assoc-acdm               12
3       44       Private  160323  Some-college               10
4       18           NaN  103497  Some-college               10
…     …            …      …           …              …
48837   27       Private  257302    Assoc-acdm               12
48838   40       Private  154374       HS-grad                9
48839   58       Private  151910       HS-grad                9
48840   22       Private  201490       HS-grad                9
48841   52  Self-emp-inc  287927       HS-grad                9

       marital-status         occupation relationship   race  gender  \
0                   2  Machine-op-inspct    Own-child  Black       1
1                   0    Farming-fishing      Husband  White       1
2                   0    Protective-serv      Husband  White       1
3                   0  Machine-op-inspct      Husband  Black       1
4                   2                NaN    Own-child  White       0
…                 …                 …           …     …      …
48837               0       Tech-support         Wife  White       0
48838               0  Machine-op-inspct      Husband  White       1
48839               3       Adm-clerical    Unmarried  White       0
```

```
48840                2       Adm-clerical    Own-child  White        1
48841                0     Exec-managerial        Wife  White        0

       capital-gain  capital-loss  hours-per-week native-country  income
0                 0             0              40  United-States       0
1                 0             0              50  United-States       0
2                 0             0              40  United-States       1
3              7688             0              40  United-States       1
4                 0             0              30  United-States       0
...             ...           ...             ...            ...     ...
48837             0             0              38  United-States       0
48838             0             0              40  United-States       1
48839             0             0              40  United-States       0
48840             0             0              20  United-States       0
48841         15024             0              40  United-States       1

[48842 rows x 15 columns]
```

[81]:
```
df['marital-status'].unique()
```

[81]:
```
array([2, 0, 3, 1])
```

[86]:
```
df
```

[86]:
```
       age     workclass  fnlwgt          education  educational-num  \
0       25       Private  226802            dropout                7
1       38       Private   89814           HighGrad                9
2       28     Local-gov  336951   CommunityCollege               12
3       44       Private  160323   CommunityCollege               10
4       18       Private  103497   CommunityCollege               10
...    ...           ...     ...                ...              ...
48837   27       Private  257302   CommunityCollege               12
48838   40       Private  154374           HighGrad                9
48839   58       Private  151910           HighGrad                9
48840   22       Private  201490           HighGrad                9
48841   52  Self-emp-inc  287927           HighGrad                9

       marital-status         occupation relationship   race  gender  \
0                   2  Machine-op-inspct    Own-child  Black       1
1                   0    Farming-fishing      Husband  White       1
2                   0    Protective-serv      Husband  White       1
3                   0  Machine-op-inspct      Husband  Black       1
4                   2     Prof-specialty    Own-child  White       0
...               ...                ...          ...    ...     ...
48837               0       Tech-support         Wife  White       0
48838               0  Machine-op-inspct      Husband  White       1
48839               3       Adm-clerical    Unmarried  White       0
```

```
48840               2       Adm-clerical    Own-child  White       1
48841               0     Exec-managerial        Wife  White       0

       capital-gain  capital-loss  hours-per-week native-country  income
0                 0             0              40  United-States       0
1                 0             0              50  United-States       0
2                 0             0              40  United-States       1
3              7688             0              40  United-States       1
4                 0             0              30  United-States       0
…               …             …              …             …     …
48837             0             0              38  United-States       0
48838             0             0              40  United-States       1
48839             0             0              40  United-States       0
48840             0             0              20  United-States       0
48841         15024             0              40  United-States       1

[48842 rows x 15 columns]
```

[88]:
```python
df['education'] = df['education'].replace('Preschool','dropout')
df['education'] = df['education'].replace('10th','dropout')
df['education'] = df['education'].replace('11th','dropout')
df['education'] = df['education'].replace('12th','dropout')
df['education'] = df['education'].replace('1st-4th','dropout')
df['education'] = df['education'].replace('5th-6th','dropout')
df['education'] = df['education'].replace('7th-8th','dropout')
df['education'] = df['education'].replace('9th','dropout')
df['education'] = df['education'].replace('HS-grad','HighGrad')
df['education'] = df['education'].replace('HS-Grad','HighGrad')
df['education'] = df['education'].replace('Some-college','CommunityCollege')
df['education'] = df['education'].replace('Assoc-acdm','CommunityCollege')
df['education'] = df['education'].replace('Assoc-voc','CommunityCollege')
df['education'] = df['education'].replace('Bachelors','Bachelors')
df['education'] = df['education'].replace('Masters','Masters')
df['education'] = df['education'].replace('Prof-school','Masters')
df['education'] = df['education'].replace('Prof-Doctorate','Doctorate')
```

[89]:
```python
df
```

[89]:
```
       age     workclass  fnlwgt          education  educational-num  \
0       25       Private  226802            dropout                7
1       38       Private   89814           HighGrad                9
2       28     Local-gov  336951   CommunityCollege               12
3       44       Private  160323   CommunityCollege               10
4       18       Private  103497   CommunityCollege               10
…     …           …     …                 …             …
48837   27       Private  257302   CommunityCollege               12
48838   40       Private  154374           HighGrad                9
```

```
48839   58        Private  151910         HighGrad                9
48840   22        Private  201490         HighGrad                9
48841   52   Self-emp-inc  287927         HighGrad                9

       marital-status         occupation relationship   race  gender  \
0                   2  Machine-op-inspct    Own-child  Black       1
1                   0    Farming-fishing      Husband  White       1
2                   0    Protective-serv      Husband  White       1
3                   0  Machine-op-inspct      Husband  Black       1
4                   2     Prof-specialty    Own-child  White       0
…                 …                  …          …     …    …
48837               0       Tech-support         Wife  White       0
48838               0  Machine-op-inspct      Husband  White       1
48839               3       Adm-clerical    Unmarried  White       0
48840               2       Adm-clerical    Own-child  White       1
48841               0    Exec-managerial         Wife  White       0

       capital-gain  capital-loss  hours-per-week native-country  income
0                 0             0              40  United-States       0
1                 0             0              50  United-States       0
2                 0             0              40  United-States       1
3              7688             0              40  United-States       1
4                 0             0              30  United-States       0
…               …           …             …            …    …
48837             0             0              38  United-States       0
48838             0             0              40  United-States       1
48839             0             0              40  United-States       0
48840             0             0              20  United-States       0
48841         15024             0              40  United-States       1

[48842 rows x 15 columns]
```

[93]:
```python
df['workclass']=lbl_encoder.fit_transform(df['workclass'])
df['occupation']=lbl_encoder.fit_transform(df['occupation'])
df['relationship']=lbl_encoder.fit_transform(df['relationship'])
df['race']=lbl_encoder.fit_transform(df['race'])
df['native-country']=lbl_encoder.fit_transform(df['native-country'])
df['education']=lbl_encoder.fit_transform(df['education'])
```

[94]:
```python
df.head()
```

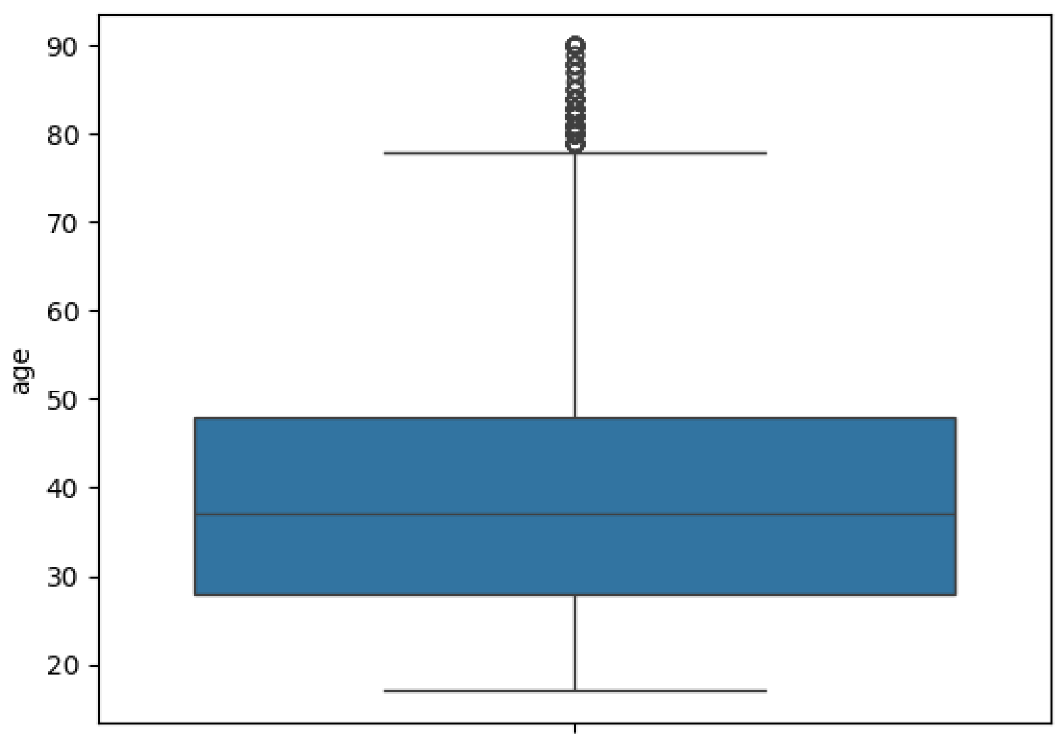
[94]:
```
   age  workclass  fnlwgt  education  educational-num  marital-status  \
0   25          3  226802          5                7               2
1   38          3   89814          3                9               0
2   28          1  336951          1               12               0
3   44          3  160323          1               10               0
4   18          3  103497          1               10               2
```

```
   occupation  relationship  race  gender  capital-gain  capital-loss  \
0           6             3     2       1             0             0
1           4             0     4       1             0             0
2          10             0     4       1             0             0
3           6             0     2       1          7688             0
4           9             3     4       0             0             0

   hours-per-week  native-country  income
0              40              38       0
1              50              38       0
2              40              38       1
3              40              38       1
4              30              38       0
```

[95]:
```
#outlier detection:-
# 1)boxplot:-  2)scatter plot 3) z score 4) inter quartile range
import seaborn as sns
sns.boxplot(df['age'])
```

[95]: <Axes: ylabel='age'>



[96]:
```
print(df['age'].unique())
```

```
[25 38 28 44 18 34 29 63 24 55 65 36 26 58 48 43 20 37 40 72 45 22 23 54
 32 46 56 17 39 52 21 42 33 30 47 41 19 69 50 31 59 49 51 27 57 61 64 79
 73 53 77 80 62 35 68 66 75 60 67 71 70 90 81 74 78 82 83 85 76 84 89 88
 87 86]
```

[97]:
```
print(np.where(df['age']>78))
```

```
(array([  193,   234,   899,   926,   951,  1079,  1398,  1834,  2085,
        2290,  2982,  3496,  3668,  4455,  4646,  4658,  6402,  6577,
        6757,  6915,  6959,  6976,  6979,  7160,  7170,  7414,  7419,
        7539,  7547,  7937,  8206,  8313,  8427,  8955,  8982,  9018,
        9038,  9081,  9279,  9769,  9888, 10039, 10199, 10223, 10735,
       11289, 11328, 11410, 11837, 11871, 11881, 11940, 12060, 12229,
       12446, 13025, 13958, 14033, 14263, 14299, 14431, 14568, 14591,
       14740, 15088, 15098, 15408, 15934, 15963, 16003, 16106, 16148,
       16251, 16355, 16503, 16711, 17199, 17321, 17449, 18216, 18584,
       19035, 19172, 19187, 19492, 19619, 19818, 20058, 20244, 20351,
       20390, 21001, 21115, 21385, 21553, 21572, 21651, 21687, 22281,
       22454, 22495, 22513, 22720, 22905, 23029, 23762, 24001, 24153,
       24457, 24662, 24712, 24803, 24975, 25087, 25244, 25254, 25752,
       26405, 26491, 26826, 27380, 27519, 27793, 27813, 28012, 28277,
       28732, 28773, 29111, 29256, 29306, 29307, 29576, 29977, 30209,
       30385, 30440, 30885, 30992, 31037, 31184, 31637, 31943, 32173,
       32583, 32804, 33043, 33182, 33890, 34318, 34422, 34553, 34558,
       34694, 34841, 35006, 35113, 35326, 35453, 35461, 35493, 35770,
       35776, 35796, 35970, 36028, 36109, 36530, 36702, 36744, 36763,
       36764, 36891, 37107, 37161, 37234, 37624, 37782, 38093, 38116,
       38501, 38762, 39176, 39179, 39740, 40181, 40308, 40324, 40519,
       40561, 40676, 40841, 41444, 41584, 41678, 42293, 42523, 43012,
       44076, 44457, 44744, 45002, 45229, 45875, 46005, 47311, 47713,
       47977, 48095, 48117, 48136, 48558, 48648, 48740, 48775, 48806],
      dtype=int64),)
```

[98]:
```
sorted_df=df.sort_values(by=['age'],ascending=True)
```

[99]:
```
sorted_df
```

[99]:

	age	workclass	fnlwgt	education	educational-num	marital-status	\
32598	17	3	133449	5	5	2	
29817	17	5	181317	5	6	2	
36580	17	3	147339	5	6	2	
26409	17	3	186677	5	7	2	
19520	17	3	110998	1	10	2	
…	…	…	…	…	…	…	
12446	90	3	347074	1	10	2	
19172	90	3	171956	1	10	1	
8982	90	3	225063	3	9	0	

```
28277   90          3   40388          0              13               2
899     90          3  149069          1              12               0

       occupation  relationship  race  gender  capital-gain  capital-loss  \
32598           7             3     2       1             0             0
29817           4             3     4       1             0             0
36580           9             3     3       0             0             0
26409           5             3     4       1             0             0
19520           9             3     1       0             0             0
…             …           …   …     …           …             …
12446           0             3     4       0             0          1944
19172           0             3     4       0             0             0
8982            2             0     1       1             0             0
28277           3             1     4       1             0             0
899            11             0     4       1             0          1825

       hours-per-week  native-country  income
32598              26              38       0
29817              35              38       0
36580              15              38       0
26409              12              38       0
19520              40              29       0
…               …             …   …
12446              12              38       0
19172              40              32       0
8982               40              34       0
28277              55              38       0
899                50              38       1

[48842 rows x 15 columns]
```

[100]:
```python
Q1=np.percentile(sorted_df['age'],25)
Q3=np.percentile(sorted_df['age'],75)
IQR=Q3-Q1
print(IQR)
```

```
20.0
```

[104]:
```python
lwr_bound=Q1-(1.5*IQR)
upr_bound=Q3+(1.5*IQR)
```

[105]:
```python
print("min:", lwr_bound, "Max:", upr_bound)
```

```
min: -2.0 Max: 78.0
```

[110]:
```python
#counting the number of outliers
outliers=[]
```

```
for i in df['age']:
    if(i<lwr_bound or i>upr_bound):
        outliers.append(i)


print("No. of outliers:",len(outliers))
print(outliers)
```

```
No. of outliers: 216
[79, 80, 90, 79, 80, 81, 82, 83, 81, 85, 80, 90, 81, 84, 81, 89, 81, 83, 81, 82,
80, 90, 81, 83, 80, 90, 90, 84, 80, 80, 80, 81, 90, 85, 90, 81, 81, 80, 80, 79,
81, 80, 88, 87, 90, 79, 83, 79, 80, 90, 79, 79, 81, 81, 90, 82, 90, 87, 81, 88,
80, 81, 80, 81, 90, 88, 89, 84, 80, 80, 83, 79, 81, 79, 90, 80, 81, 90, 88, 90,
90, 80, 90, 81, 82, 79, 81, 80, 83, 90, 90, 79, 81, 90, 90, 80, 90, 90, 79, 79,
84, 90, 80, 90, 81, 83, 84, 81, 79, 85, 82, 79, 80, 90, 90, 90, 84, 80, 90, 90,
79, 84, 90, 79, 90, 90, 90, 82, 81, 90, 84, 79, 81, 82, 81, 80, 90, 80, 84, 82,
79, 90, 84, 90, 83, 79, 81, 80, 79, 80, 79, 80, 90, 90, 80, 90, 90, 81, 83, 82,
90, 90, 81, 80, 80, 90, 79, 80, 82, 85, 80, 79, 90, 81, 79, 80, 79, 81, 82, 88,
90, 82, 88, 84, 83, 79, 86, 90, 90, 82, 83, 81, 79, 90, 80, 81, 79, 84, 84, 79,
90, 80, 81, 81, 81, 90, 87, 90, 80, 80, 82, 90, 90, 85, 82, 81]
```

[111]:
```
#handling the outliers
#1)removing outliers
#2) quartile based flooring and capping
#3) mean/median imputation
median=np.median(df['age'])
print(median)
for i in outliers:
    df['age']=np.where(df['age']==i,37,df['age'])
```

```
37.0
```

[112]:
```
df
```

[112]:

	age	workclass	fnlwgt	education	educational-num	marital-status	\
0	25	3	226802	5	7	2	
1	38	3	89814	3	9	0	
2	28	1	336951	1	12	0	
3	44	3	160323	1	10	0	
4	18	3	103497	1	10	2	
…	…	…	…	…	…	…	
48837	27	3	257302	1	12	0	
48838	40	3	154374	3	9	0	
48839	58	3	151910	3	9	3	
48840	22	3	201490	3	9	2	
48841	52	4	287927	3	9	0	

```
       occupation  relationship  race  gender  capital-gain  capital-loss  \
0               6             3     2       1             0             0
1               4             0     4       1             0             0
2              10             0     4       1             0             0
3               6             0     2       1          7688             0
4               9             3     4       0             0             0
…             …             …    …     …           …           …
48837          12             5     4       0             0             0
48838           6             0     4       1             0             0
48839           0             4     4       0             0             0
48840           0             3     4       1             0             0
48841           3             5     4       0         15024             0

       hours-per-week  native-country  income
0                  40              38       0
1                  50              38       0
2                  40              38       1
3                  40              38       1
4                  30              38       0
…                 …             …     …
48837              38              38       0
48838              40              38       1
48839              40              38       0
48840              20              38       0
48841              40              38       1

[48842 rows x 15 columns]
```

[114]:
```
df['age'].sort_values().unique()
```

[114]:
```
array([17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
       34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50,
       51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67,
       68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78], dtype=int64)
```

[115]:
```
#data sampling:-
df['income'].value_counts()
```

[115]:
```
income
0    37155
1    11687
Name: count, dtype: int64
```

[122]:
```
#random sampling
lt_fifty_k=df[df['income']==0]
gt_fifty_k=df[df['income']==1]
```

[123]:
```
print("<=50k:", lt_fifty_k.shape)
print(">50k:",gt_fifty_k.shape)
```

```
<=50k: (37155, 15)
>50k: (11687, 15)
```

[124]:
```
no_sample=lt_fifty_k.sample(n=11681)
```

[126]:
```
no_sample.shape
```

[126]:
```
(11681, 15)
```

[127]:
```
sample_df=pd.concat([no_sample,gt_fifty_k],axis=0)
```

[129]:
```
sample_df.shape
```

[129]:
```
(23368, 15)
```

[131]:
```
sample_df['income'].value_counts()
```

[131]:
```
income
1    11687
0    11681
Name: count, dtype: int64
```

[133]:
```
sample_df
```

[133]:
```
       age  workclass  fnlwgt  education  educational-num  marital-status  \
35042   41          3  126076          1               10               1
9139    24          3  206974          0               13               2
46229   37          6   74163          0               13               2
32499   19          3  178147          1               10               2
2484    60          3  178764          3                9               0
…     …         …       …          …                …               …
48820   71          3  287372          2               16               0
48826   39          1  111499          1               12               0
48835   53          3  321865          4               14               0
48838   40          3  154374          3                9               0
48841   52          4  287927          3                9               0

       occupation  relationship  race  gender  capital-gain  capital-loss  \
35042           2             1     4       0             0             0
9139            0             3     4       0             0             0
46229           9             1     4       0             0             0
32499           5             3     4       1             0             0
2484            9             0     4       1             0             0
…           …             …     …       …             …             …
```

```
48820          9          0     4     1           0          0
48826          0          5     4     0           0          0
48835          3          0     4     1           0          0
48838          6          0     4     1           0          0
48841          3          5     4     0       15024          0

       hours-per-week  native-country  income
35042              50              38       0
9139               40              38       0
46229              40              38       0
32499              10              38       0
2484               25              38       0
…                   …               …       …
48820              10              38       1
48826              20              38       1
48835              40              38       1
48838              40              38       1
48841              40              38       1

[23368 rows x 15 columns]
```

[134]:

```python
X=df.drop('income',axis=1)
y=df['income']
```

[135]:

```python
print("Shape of X:", X.shape)
print("shape of y:",y.shape)
```

```
Shape of X: (48842, 14)
shape of y: (48842,)
```

[136]:

```python
#selecting the feature
df.corr()
```

[136]:

```
                      age  workclass    fnlwgt  education  educational-num  \
age              1.000000   0.044513 -0.073686   0.063756         0.036628
workclass        0.044513   1.000000 -0.026519   0.011359         0.007333
fnlwgt          -0.073686  -0.026519  1.000000   0.019273        -0.038761
education        0.063756   0.011359  0.019273   1.000000        -0.605925
educational-num  0.036628   0.007333 -0.038761  -0.605925         1.000000
marital-status  -0.345922  -0.054778  0.022223   0.003764        -0.077434
occupation      -0.002124   0.009841 -0.002253   0.006570         0.072706
relationship    -0.265535  -0.056073  0.009092   0.021134        -0.090534
race             0.027786   0.053923 -0.027062  -0.020241         0.029239
gender           0.089214   0.066672  0.027739   0.033259         0.009328
capital-gain     0.077980   0.031558 -0.003706  -0.006323         0.125146
capital-loss     0.056789   0.004168 -0.004366  -0.024336         0.080972
hours-per-week   0.088343   0.042845 -0.013519  -0.060260         0.143689
```

native-country	-0.002536	-0.004829	-0.058534	-0.082127	0.090137
income	0.238385	-0.000511	-0.006339	-0.134551	0.332613

	marital-status	occupation	relationship	race	gender	\
age	-0.345922	-0.002124	-0.265535	0.027786	0.089214	
workclass	-0.054778	0.009841	-0.056073	0.053923	0.066672	
fnlwgt	0.022223	-0.002253	0.009092	-0.027062	0.027739	
education	0.003764	0.006570	0.021134	-0.020241	0.033259	
educational-num	-0.077434	0.072706	-0.090534	0.029239	0.009328	
marital-status	1.000000	0.003720	0.439632	-0.075040	-0.370274	
occupation	0.003720	1.000000	-0.034964	-0.005210	0.042579	
relationship	0.439632	-0.034964	1.000000	-0.117041	-0.579797	
race	-0.075040	-0.005210	-0.117041	1.000000	0.086734	
gender	-0.370274	0.042579	-0.579797	0.086734	1.000000	
capital-gain	-0.077956	0.014518	-0.056510	0.011581	0.047094	
capital-loss	-0.067888	0.011082	-0.057201	0.018595	0.045480	
hours-per-week	-0.244961	-0.015550	-0.250400	0.039694	0.228560	
native-country	0.019883	-0.001577	-0.006999	0.117553	-0.002453	
income	-0.407109	0.032550	-0.253214	0.070934	0.214628	

	capital-gain	capital-loss	hours-per-week	native-country	\
age	0.077980	0.056789	0.088343	-0.002536	
workclass	0.031558	0.004168	0.042845	-0.004829	
fnlwgt	-0.003706	-0.004366	-0.013519	-0.058534	
education	-0.006323	-0.024336	-0.060260	-0.082127	
educational-num	0.125146	0.080972	0.143689	0.090137	
marital-status	-0.077956	-0.067888	-0.244961	0.019883	
occupation	0.014518	0.011082	-0.015550	-0.001577	
relationship	-0.056510	-0.057201	-0.250400	-0.006999	
race	0.011581	0.018595	0.039694	0.117553	
gender	0.047094	0.045480	0.228560	-0.002453	
capital-gain	1.000000	-0.031441	0.082157	0.007919	
capital-loss	-0.031441	1.000000	0.054467	0.006523	
hours-per-week	0.082157	0.054467	1.000000	0.006497	
native-country	0.007919	0.006523	0.006497	1.000000	
income	0.223013	0.147554	0.227687	0.020375	

	income
age	0.238385
workclass	-0.000511
fnlwgt	-0.006339
education	-0.134551
educational-num	0.332613
marital-status	-0.407109
occupation	0.032550
relationship	-0.253214
race	0.070934

```
gender            0.214628
capital-gain      0.223013
capital-loss      0.147554
hours-per-week    0.227687
native-country    0.020375
income            1.000000
```

[138]:
```
from sklearn.feature_selection import mutual_info_classif
#determine the mutual information
mutual_info=mutual_info_classif(X,y)
mutual_info
```

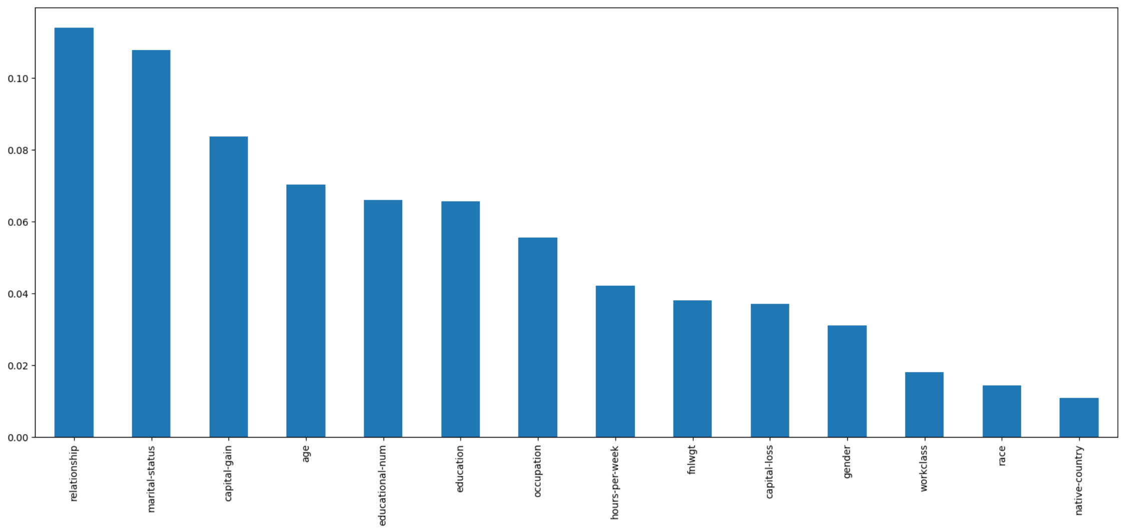
[138]:
```
array([0.0703883 , 0.01808248, 0.03802097, 0.06557895, 0.06602875,
       0.10768541, 0.05557049, 0.11394699, 0.01430229, 0.03113962,
       0.08367703, 0.03698271, 0.04218606, 0.01081403])
```

[139]:
```
mutual_info=pd.Series(mutual_info)
mutual_info.index=X.columns
mutual_info.sort_values(ascending=False)
```

[139]:
```
relationship       0.113947
marital-status     0.107685
capital-gain       0.083677
age                0.070388
educational-num    0.066029
education          0.065579
occupation         0.055570
hours-per-week     0.042186
fnlwgt             0.038021
capital-loss       0.036983
gender             0.031140
workclass          0.018082
race               0.014302
native-country     0.010814
dtype: float64
```

[140]:
```
mutual_info.sort_values(ascending=False).plot.bar(figsize=(20,8))
```

[140]: <Axes: >



[142]:
```
X=df.
↪drop(['workclass','race','native-country','gender','capital-loss','income'],axis=1)
```

[143]:
```
X
```

[143]:
```
       age  fnlwgt  education  educational-num  marital-status  occupation  \
0       25  226802          5                7               2           6
1       38   89814          3                9               0           4
2       28  336951          1               12               0          10
3       44  160323          1               10               0           6
4       18  103497          1               10               2           9
…      …     …       …               …                …            …
48837   27  257302          1               12               0          12
48838   40  154374          3                9               0           6
48839   58  151910          3                9               3           0
48840   22  201490          3                9               2           0
48841   52  287927          3                9               0           3

       relationship  capital-gain  hours-per-week
0                 3             0              40
1                 0             0              50
2                 0             0              40
3                 0          7688              40
4                 3             0              30
…              …             …              …
48837             5             0              38
48838             0             0              40
48839             4             0              40
48840             3             0              20
```

```
48841                5          15024             40

[48842 rows x 9 columns]
```

[146]:
```
from sklearn.model_selection import train_test_split
X_train, X_test,y_train, y_test=train_test_split(X,y,test_size=0.
 ↪3,random_state=42,shuffle=True)
```

[148]:
```
print('X_Training Shape:',X_train.shape)
print('X_Testing Shape:',X_test.shape)
print('Y_Training Shape:',y_train.shape)
print('y test Shape:',y_test.shape)
```

```
X_Training Shape: (34189, 9)
X_Testing Shape: (14653, 9)
Y_Training Shape: (34189,)
y test Shape: (14653,)
```

[ ]:
```
```