knn1

November 21, 2024

[1]:
```
!pip install -q autoviz
!pip install -q -U --pre pycaret
```

```
ERROR: Could not install packages due to an OSError: [WinError 5] Access is
denied: 'C:\\Users\\ANUSHA\\AppData\\Roaming\\Python\\Python39\\site-
packages\\~cipy.libs\\libopenblas_v0.3.27--3aa239bc726cfb0bd8e5330d8d4c15c6.dll'
Check the permissions.
```

[24]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
from statsmodels.stats.outliers_influence import variance_inflation_factor
from pycaret.regression import setup, compare_models, create_model,␣
↪evaluate_model
from sklearn.model_selection import cross_val_score
```

[25]:
```
df = pd.read_csv('C:/Users/ANUSHA/Downloads/KNN.csv')
```

[26]:
```
df.head()
```

[26]:
```
   Gender  Age  Height (cm)  Weight (kg)          Occupation  \
0    male   32          175           70   Software Engineer
1    male   25          182           85  Sales Representative
2  female   41          160           62              Doctor
3    male   38          178           79              Lawyer
4  female   29          165           58    Graphic Designer

    Education Level Marital Status  Income (USD) Favorite Color  \
0   Master's Degree        Married         75000           Blue
1 Bachelor's Degree         Single         45000          Green
2  Doctorate Degree        Married        120000         Purple
3 Bachelor's Degree         Single         90000            Red
4 Associate's Degree        Single         35000         Yellow
```

```
   Unnamed: 9
0         NaN
1         NaN
2         NaN
3         NaN
4         NaN
```

[27]:
```
df.shape
```

[27]:
```
(131, 10)
```

[28]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 131 entries, 0 to 130
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0    Gender           131 non-null    object
 1    Age              131 non-null    int64
 2    Height (cm)      131 non-null    int64
 3    Weight (kg)      131 non-null    int64
 4    Occupation       131 non-null    object
 5    Education Level  131 non-null    object
 6    Marital Status   131 non-null    object
 7    Income (USD)     131 non-null    int64
 8    Favorite Color   131 non-null    object
 9   Unnamed: 9       0 non-null      float64
dtypes: float64(1), int64(4), object(5)
memory usage: 10.4+ KB
```

[29]:
```
df.describe()
```

[29]:

	Age	Height (cm)	Weight (kg)	Income (USD)	Unnamed: 9
count	131.000000	131.000000	131.000000	131.000000	0.0
mean	34.564885	173.198473	71.458015	93206.106870	NaN
std	5.984723	8.045467	12.648052	74045.382919	NaN
min	24.000000	160.000000	50.000000	30000.000000	NaN
25%	29.000000	166.000000	60.000000	55000.000000	NaN
50%	34.000000	175.000000	75.000000	75000.000000	NaN
75%	39.000000	180.500000	83.000000	100000.000000	NaN
max	52.000000	190.000000	94.000000	500000.000000	NaN

[30]:
```
df.drop(['Unnamed: 9'], axis = 1, inplace = True)
df
```

[30]:

	Gender	Age	Height (cm)	Weight (kg)	Occupation	\
0	male	32	175	70	Software Engineer	
1	male	25	182	85	Sales Representative	
2	female	41	160	62	Doctor	
3	male	38	178	79	Lawyer	
4	female	29	165	58	Graphic Designer	
5	male	45	190	92	Business Consultant	
6	female	27	163	55	Marketing Specialist	
7	male	52	179	83	CEO	
8	female	31	168	61	Project Manager	
9	male	36	177	76	Engineer	
10	female	24	162	53	Accountant	
11	male	44	183	87	Architect	
12	female	28	166	60	Nurse	
13	male	29	181	84	Analyst	
14	female	33	170	65	Teacher	
15	male	37	176	78	IT Manager	
16	female	26	169	59	Writer	
17	male	28	182	75	Engineer	
18	male	33	178	82	Teacher	
19	female	44	160	58	Doctor	
20	male	29	176	74	Sales Representative	
21	female	31	165	63	Marketing Specialist	
22	male	40	187	90	Business Analyst	
23	female	27	163	56	Project Manager	
24	male	47	181	85	CEO	
25	female	35	170	65	Graphic Designer	
26	male	42	175	80	Architect	
27	female	26	160	53	Accountant	
28	male	49	183	92	Lawyer	
29	female	30	168	61	Nurse	
30	male	35	181	84	Analyst	
31	female	38	172	68	Teacher	
32	male	31	175	78	IT Manager	
33	female	29	166	59	Writer	
34	male	32	175	78	Engineer	
35	female	27	168	61	Marketing Specialist	
36	male	41	180	85	Business Analyst	
37	female	36	163	57	Graphic Designer	
38	male	29	183	81	Sales Representative	
39	female	33	170	65	Doctor	
40	male	45	187	92	CEO	
41	female	28	160	50	Writer	
42	male	39	179	83	IT Manager	
43	female	32	165	58	Accountant	
44	male	44	183	90	Lawyer	
45	female	26	162	54	Teacher	

46	male	36	176	76	Analyst
47	female	29	167	63	Project Manager
48	male	42	181	87	Architect
49	female	30	169	62	Nurse
50	male	34	177	79	Teacher
51	male	35	178	80	Engineer
52	female	28	165	60	Teacher
53	male	42	185	90	Doctor
54	female	31	163	55	Graphic Designer
55	male	30	182	83	IT Manager
56	female	36	170	65	Sales Representative
57	male	44	188	92	Lawyer
58	female	29	167	60	Marketing Specialist
59	male	37	179	81	Project Manager
60	female	26	162	55	Writer
61	male	43	183	85	Architect
62	female	34	168	63	Nurse
63	male	36	174	75	Business Analyst
64	female	27	166	56	Accountant
65	male	41	180	86	CEO
66	female	30	170	62	Teacher
67	male	38	175	77	Analyst
68	female	29	164	56	Doctor
69	male	40	182	88	Engineer
70	female	33	169	63	Marketing Specialist
71	male	39	181	82	IT Manager
72	female	32	168	60	Writer
73	male	45	186	94	Lawyer
74	female	28	163	53	Graphic Designer
75	male	34	177	80	Sales Representative
76	female	31	166	58	Teacher
77	male	42	184	87	Architect
78	female	30	170	64	Nurse
79	male	37	179	79	Project Manager
80	male	37	175	76	Engineer
81	female	27	160	54	Teacher
82	male	44	182	88	Doctor
83	female	32	168	63	Graphic Designer
84	male	39	178	81	IT Manager
85	female	29	165	57	Marketing Specialist
86	male	43	183	85	Lawyer
87	female	31	170	63	Nurse
88	male	38	176	79	Project Manager
89	female	28	162	55	Writer
90	male	41	181	86	Architect
91	female	33	170	64	Accountant
92	male	36	179	78	Business Analyst

93	female	30	163	58	Teacher
94	male	42	184	90	CEO
95	female	34	168	62	Marketing Specialist
96	male	35	177	80	Analyst
97	female	29	166	58	Doctor
98	male	40	183	88	Engineer
99	female	31	167	60	Writer
100	male	43	185	84	Lawyer
101	female	32	168	63	Nurse
102	male	37	179	80	Project Manager
103	female	28	163	56	Graphic Designer
104	male	34	176	78	Sales Representative
105	male	32	178	78	Engineer
106	female	27	162	56	Teacher
107	male	44	182	88	Doctor
108	female	35	168	64	Graphic Designer
109	male	38	179	80	IT Manager
110	female	30	164	58	Marketing Specialist
111	male	42	183	87	Lawyer
112	female	29	160	55	Nurse
113	male	36	175	76	Project Manager
114	female	31	166	60	Writer
115	male	41	184	85	Architect
116	female	33	169	62	Accountant
117	male	35	178	79	Business Analyst
118	female	28	162	55	Teacher
119	male	40	183	86	CEO
120	female	34	167	61	Marketing Specialist
121	male	37	180	82	Analyst
122	female	29	165	57	Doctor
123	male	40	182	88	Engineer
124	female	31	168	61	Writer
125	male	43	184	85	Lawyer
126	female	32	170	64	Nurse
127	male	38	176	79	Project Manager
128	female	27	162	55	Graphic Designer
129	male	33	175	77	Sales Representative
130	female	29	164	57	Software Developer

	Education Level	Marital Status	Income (USD)	Favorite Color
0	Master's Degree	Married	75000	Blue
1	Bachelor's Degree	Single	45000	Green
2	Doctorate Degree	Married	120000	Purple
3	Bachelor's Degree	Single	90000	Red
4	Associate's Degree	Single	35000	Yellow
5	Master's Degree	Divorced	110000	Black
6	Bachelor's Degree	Single	50000	Pink

7	Doctorate Degree	Married	500000	Blue
8	Bachelor's Degree	Married	80000	Green
9	Master's Degree	Married	95000	Orange
10	Bachelor's Degree	Single	40000	Blue
11	Bachelor's Degree	Widowed	120000	Grey
12	Associate's Degree	Married	55000	Purple
13	Bachelor's Degree	Single	60000	Red
14	Master's Degree	Married	65000	Yellow
15	Bachelor's Degree	Married	85000	Green
16	Bachelor's Degree	Single	30000	Pink
17	Bachelor's Degree	Married	80000	Blue
18	Master's Degree	Single	45000	Green
19	Doctorate Degree	Married	150000	Purple
20	Bachelor's Degree	Single	55000	Red
21	Bachelor's Degree	Single	50000	Yellow
22	Master's Degree	Divorced	120000	Black
23	Bachelor's Degree	Single	60000	Pink
24	Doctorate Degree	Married	500000	Blue
25	Associate's Degree	Married	70000	Green
26	Bachelor's Degree	Married	100000	Orange
27	Bachelor's Degree	Single	40000	Blue
28	Bachelor's Degree	Widowed	150000	Grey
29	Associate's Degree	Married	60000	Purple
30	Bachelor's Degree	Single	70000	Red
31	Master's Degree	Married	75000	Yellow
32	Bachelor's Degree	Married	90000	Green
33	Bachelor's Degree	Single	35000	Pink
34	Master's Degree	Married	90000	Blue
35	Bachelor's Degree	Single	55000	Pink
36	Bachelor's Degree	Divorced	110000	Green
37	Associate's Degree	Married	65000	Purple
38	Bachelor's Degree	Single	50000	Black
39	Doctorate Degree	Married	130000	Red
40	Doctorate Degree	Married	500000	Yellow
41	Bachelor's Degree	Single	40000	Grey
42	Bachelor's Degree	Married	95000	Orange
43	Bachelor's Degree	Single	60000	Blue
44	Master's Degree	Married	180000	Pink
45	Bachelor's Degree	Single	35000	Green
46	Bachelor's Degree	Divorced	75000	Purple
47	Master's Degree	Single	55000	Red
48	Bachelor's Degree	Married	120000	Yellow
49	Associate's Degree	Married	60000	Black
50	Bachelor's Degree	Single	60000	Grey
51	Master's Degree	Single	90000	Blue
52	Bachelor's Degree	Married	55000	Green
53	Doctorate Degree	Married	120000	Red

54	Associate's Degree	Single	65000	Orange
55	Bachelor's Degree	Single	95000	Purple
56	Bachelor's Degree	Married	75000	Yellow
57	Master's Degree	Divorced	150000	Black
58	Bachelor's Degree	Single	60000	Grey
59	Bachelor's Degree	Married	85000	Pink
60	Bachelor's Degree	Single	40000	Blue
61	Bachelor's Degree	Married	110000	Green
62	Associate's Degree	Married	65000	Red
63	Bachelor's Degree	Married	95000	Purple
64	Bachelor's Degree	Single	55000	Yellow
65	Doctorate Degree	Married	250000	Orange
66	Bachelor's Degree	Single	60000	Blue
67	Bachelor's Degree	Married	80000	Green
68	Doctorate Degree	Single	120000	Red
69	Master's Degree	Married	100000	Purple
70	Bachelor's Degree	Married	70000	Yellow
71	Bachelor's Degree	Single	90000	Black
72	Bachelor's Degree	Single	45000	Grey
73	Master's Degree	Married	150000	Blue
74	Associate's Degree	Single	55000	Green
75	Bachelor's Degree	Single	50000	Red
76	Bachelor's Degree	Married	65000	Orange
77	Bachelor's Degree	Married	130000	Purple
78	Associate's Degree	Single	55000	Yellow
79	Bachelor's Degree	Married	95000	Black
80	Bachelor's Degree	Married	80000	Blue
81	Bachelor's Degree	Single	45000	Purple
82	Doctorate Degree	Married	120000	Green
83	Associate's Degree	Single	60000	Yellow
84	Bachelor's Degree	Married	100000	Orange
85	Bachelor's Degree	Single	65000	Red
86	Master's Degree	Married	150000	Purple
87	Associate's Degree	Single	55000	Grey
88	Bachelor's Degree	Married	90000	Black
89	Bachelor's Degree	Single	50000	Green
90	Bachelor's Degree	Married	110000	Blue
91	Bachelor's Degree	Single	70000	Red
92	Bachelor's Degree	Married	85000	Orange
93	Bachelor's Degree	Single	55000	Yellow
94	Doctorate Degree	Married	250000	Purple
95	Bachelor's Degree	Single	75000	Green
96	Bachelor's Degree	Married	70000	Blue
97	Doctorate Degree	Single	120000	Red
98	Master's Degree	Married	100000	Black
99	Bachelor's Degree	Single	45000	Orange
100	Master's Degree	Married	130000	Purple

```
101   Associate's Degree      Single      60000        Yellow
102    Bachelor's Degree     Married      90000         Green
103   Associate's Degree      Single      55000          Blue
104    Bachelor's Degree     Married      80000          Grey
105    Bachelor's Degree     Married      75000          Blue
106    Bachelor's Degree      Single      50000         Green
107     Doctorate Degree     Married     120000           Red
108   Associate's Degree      Single      65000        Purple
109    Bachelor's Degree     Married      90000        Orange
110    Bachelor's Degree      Single      55000         Black
111      Master's Degree     Married     150000          Blue
112   Associate's Degree      Single      45000        Yellow
113    Bachelor's Degree     Married      80000         Green
114    Bachelor's Degree      Single      55000           Red
115    Bachelor's Degree     Married     110000        Purple
116    Bachelor's Degree      Single      70000          Blue
117    Bachelor's Degree     Married      85000         Green
118    Bachelor's Degree      Single      50000        Orange
119     Doctorate Degree     Married     250000         Black
120    Bachelor's Degree      Single      75000           Red
121    Bachelor's Degree     Married      70000        Yellow
122     Doctorate Degree      Single     120000         Green
123      Master's Degree     Married     100000          Blue
124    Bachelor's Degree      Single      45000        Purple
125      Master's Degree     Married     130000           Red
126   Associate's Degree      Single      60000        Orange
127    Bachelor's Degree     Married      90000         Black
128   Associate's Degree      Single      55000         Green
129    Bachelor's Degree     Married      80000        Yellow
130    Bachelor's Degree      Single      65000          Blue
```

[31]:
```
cat_cols = df.select_dtypes(include=['object']).columns.tolist()
cat_cols
```

[31]:
```
[' Gender',
 ' Occupation',
 ' Education Level',
 ' Marital Status',
 ' Favorite Color']
```

[32]:
```
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()

for col in cat_cols:
    # Encode values in training set
    le.fit(df[col])
```

```
    df[col] = le.transform(df[col])
```

[33]:
```
df.isnull().any()
```

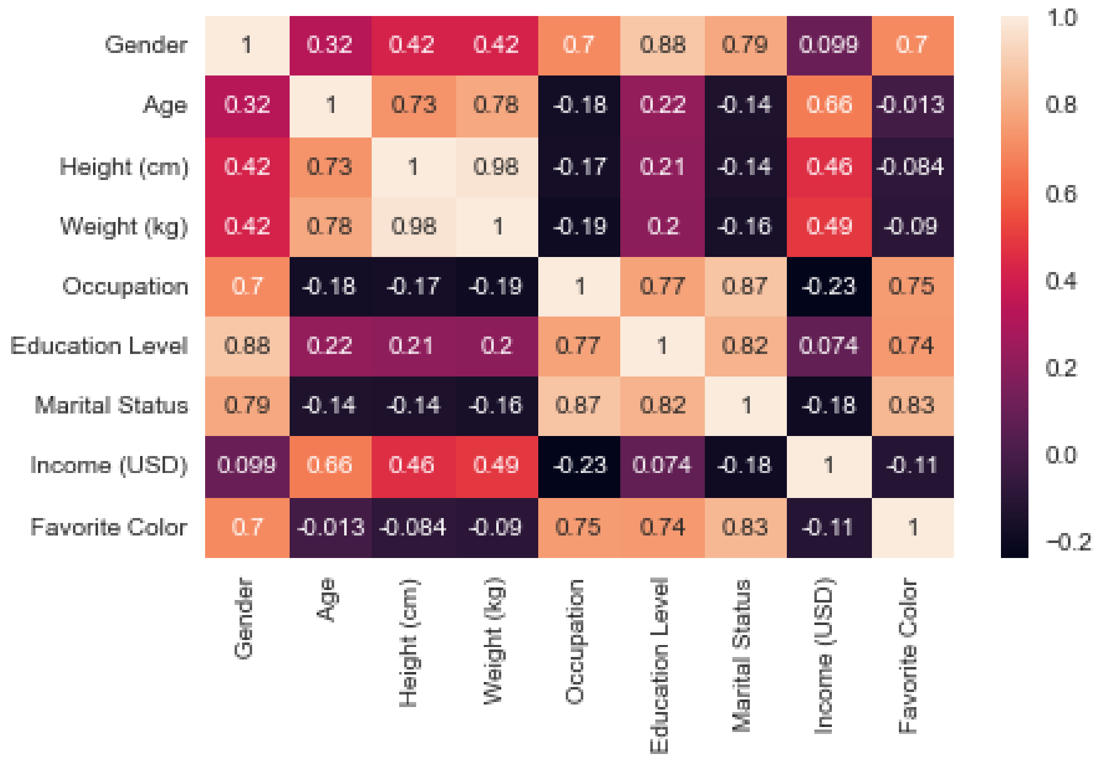
[33]:
```
Gender            False
Age               False
Height (cm)       False
Weight (kg)       False
Occupation        False
Education Level   False
Marital Status    False
Income (USD)      False
Favorite Color    False
dtype: bool
```

[34]:
```
df.corr()
```

[34]:
```
                  Gender       Age  Height (cm)  Weight (kg)  Occupation  \
Gender          1.000000  0.320236     0.417210     0.420043    0.696525
Age             0.320236  1.000000     0.726308     0.784738   -0.175900
Height (cm)     0.417210  0.726308     1.000000     0.975157   -0.167029
Weight (kg)     0.420043  0.784738     0.975157     1.000000   -0.188074
Occupation      0.696525 -0.175900    -0.167029    -0.188074    1.000000
Education Level 0.883911  0.220823     0.208090     0.200688    0.771409
Marital Status  0.785893 -0.135039    -0.141778    -0.156897    0.872901
Income (USD)    0.099271  0.662278     0.456217     0.486022   -0.234879
Favorite Color  0.696996 -0.013207    -0.084310    -0.090207    0.751068

                Education Level  Marital Status  Income (USD)  Favorite Color
Gender                 0.883911        0.785893      0.099271        0.696996
Age                    0.220823       -0.135039      0.662278       -0.013207
Height (cm)            0.208090       -0.141778      0.456217       -0.084310
Weight (kg)            0.200688       -0.156897      0.486022       -0.090207
Occupation             0.771409        0.872901     -0.234879        0.751068
Education Level        1.000000        0.820698      0.074108        0.743115
Marital Status         0.820698        1.000000     -0.183237        0.834993
Income (USD)           0.074108       -0.183237      1.000000       -0.111655
Favorite Color         0.743115        0.834993     -0.111655        1.000000
```

[35]:
```
sns.heatmap(df.corr(), annot = True)
```

[35]:
```
<Axes: >
```



[36]:
```
class_counts = df[' Gender'].value_counts()
print('Class distribution:')
print(class_counts)
```

```
Class distribution:
 Gender
3    41
2    39
1    27
0    24
Name: count, dtype: int64
```

[37]:
```
def data_cleaning_suggestions(df):
    print("Basic Information:")
    print(df.info())
    print("\nMissing Values:")
    print(df.isnull().sum())

    print("\nDuplicate Rows:")
    print(df.duplicated().sum())

    print("\nDescriptive Statistics:")
```

```
    print(df.describe())

    print("\nColumns with High Cardinality (Unique Values):")
    for col in df.columns:
        if df[col].nunique() > 50:
            print(f"  - {col}: {df[col].nunique()} unique values")

    print("\nPotential Outliers (Z-score > 3):")
    numerical_cols = df.select_dtypes(include=np.number)
    z_scores = np.abs(stats.zscore(numerical_cols))
    outliers = (z_scores > 3).sum(axis=0)
    print(outliers)

data_cleaning_suggestions(df)
for col in df.select_dtypes(include=np.number).columns:
    df[col].fillna(df[col].median(), inplace=True)
label_encoder = LabelEncoder()
for col in df.select_dtypes(include='object').columns:
    df[col] = label_encoder.fit_transform(df[col])

df.drop_duplicates(inplace=True)
numerical_cols = df.select_dtypes(include=np.number)
for col in numerical_cols.columns:
    q1 = df[col].quantile(0.25)
    q3 = df[col].quantile(0.75)
    iqr = q3 - q1
    lower_limit = q1 - 1.5 * iqr
    upper_limit = q3 + 1.5 * iqr
    df[col] = np.clip(df[col], lower_limit, upper_limit)

print("\nCleaned Dataset Preview:")
print(df.head())
```

```
Basic Information:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 131 entries, 0 to 130
Data columns (total 9 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0    Gender           131 non-null    int32
 1    Age              131 non-null    int64
 2    Height (cm)      131 non-null    int64
 3    Weight (kg)      131 non-null    int64
 4    Occupation       131 non-null    int32
 5    Education Level  131 non-null    int32
 6    Marital Status   131 non-null    int32
 7    Income (USD)     131 non-null    int64
```

```
 8    Favorite Color   131 non-null    int32
dtypes: int32(5), int64(4)
memory usage: 6.8 KB
None

Missing Values:
Gender             0
Age                0
Height (cm)        0
Weight (kg)        0
Occupation         0
Education Level    0
Marital Status     0
Income (USD)       0
Favorite Color     0
dtype: int64

Duplicate Rows:
1

Descriptive Statistics:
           Gender         Age  Height (cm)  Weight (kg)  Occupation  \
count  131.000000  131.000000   131.000000   131.000000  131.000000
mean     1.740458   34.564885   173.198473    71.458015   19.702290
std      1.092547    5.984723     8.045467    12.648052    9.995918
min      0.000000   24.000000   160.000000    50.000000    0.000000
25%      1.000000   29.000000   166.000000    60.000000   11.500000
50%      2.000000   34.000000   175.000000    75.000000   22.000000
75%      3.000000   39.000000   180.500000    83.000000   28.000000
max      3.000000   52.000000   190.000000    94.000000   34.000000

       Education Level  Marital Status   Income (USD)  Favorite Color
count       131.000000      131.000000     131.000000      131.000000
mean          3.725191        3.885496   93206.106870        9.458015
std           2.068271        2.070456   74045.382919        5.147456
min           0.000000        0.000000   30000.000000        0.000000
25%           1.000000        2.000000   55000.000000        5.500000
50%           5.000000        5.000000   75000.000000       10.000000
75%           5.000000        6.000000  100000.000000       13.500000
max           7.000000        6.000000  500000.000000       17.000000

Columns with High Cardinality (Unique Values):

Potential Outliers (Z-score > 3):
Gender             0
Age                0
Height (cm)        0
Weight (kg)        0
```

```
Occupation         0
Education Level    0
Marital Status     0
Income (USD)       3
Favorite Color     0
dtype: int64

Cleaned Dataset Preview:
   Gender  Age  Height (cm)  Weight (kg)  Occupation  Education Level  \
0       1   32          175           70          15                3
1       1   25          182           85          14                1
2       0   41          160           62           6                2
3       1   38          178           79          10                1
4       0   29          165           58           8                0

   Marital Status  Income (USD)  Favorite Color
0               1         75000               1
1               2         45000               2
2               1        120000               6
3               2         90000               7
4               2         35000               8
```

[38]:

```
data_cleaning_suggestions(df)
```

```
Basic Information:
<class 'pandas.core.frame.DataFrame'>
Index: 130 entries, 0 to 130
Data columns (total 9 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Gender           130 non-null    int32
 1   Age              130 non-null    int64
 2   Height (cm)      130 non-null    int64
 3   Weight (kg)      130 non-null    int64
 4   Occupation       130 non-null    int32
 5   Education Level  130 non-null    int32
 6   Marital Status   130 non-null    int32
 7   Income (USD)     130 non-null    int64
 8   Favorite Color   130 non-null    int32
dtypes: int32(5), int64(4)
memory usage: 7.6 KB
None

Missing Values:
Gender             0
Age                0
Height (cm)        0
Weight (kg)        0
```

```
Occupation         0
Education Level    0
Marital Status     0
Income (USD)       0
Favorite Color     0
dtype: int64

Duplicate Rows:
0

Descriptive Statistics:
           Gender         Age  Height (cm)  Weight (kg)  Occupation  \
count  130.000000  130.000000   130.000000   130.000000  130.000000
mean     1.730769   34.538462   173.176923    71.400000   19.623077
std      1.091109    6.000199     8.072794    12.679471    9.993227
min      0.000000   24.000000   160.000000    50.000000    0.000000
25%      1.000000   29.000000   166.000000    60.000000   11.250000
50%      2.000000   33.500000   175.000000    74.500000   22.000000
75%      3.000000   39.000000   180.750000    83.000000   28.000000
max      3.000000   52.000000   190.000000    94.000000   34.000000

       Education Level  Marital Status   Income (USD)  Favorite Color
count       130.000000      130.000000     130.000000      130.000000
mean          3.715385        3.876923   83557.692308        9.461538
std           2.073213        2.076130   35388.028163        5.167210
min           0.000000        0.000000   30000.000000        0.000000
25%           1.000000        2.000000   55000.000000        5.250000
50%           5.000000        5.000000   75000.000000       10.000000
75%           5.000000        6.000000  100000.000000       13.750000
max           7.000000        6.000000  167500.000000       17.000000

Columns with High Cardinality (Unique Values):

Potential Outliers (Z-score > 3):
Gender             0
Age                0
Height (cm)        0
Weight (kg)        0
Occupation         0
Education Level    0
Marital Status     0
Income (USD)       0
Favorite Color     0
dtype: int64
```

[39]:
```python
X = df.drop(' Gender', axis = 1)
y = df[' Gender']
```

[40]:
```python
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)
```

[41]:
```python
from imblearn.over_sampling import SMOTE

# Instantiate SMOTE
sm = SMOTE(random_state=42)
X_train_res, y_train_res = sm.fit_resample(X_train, y_train)
print('Class distribution before resampling:', y_train.value_counts())
print('Class distribution after resampling:', y_train_res.value_counts())
```

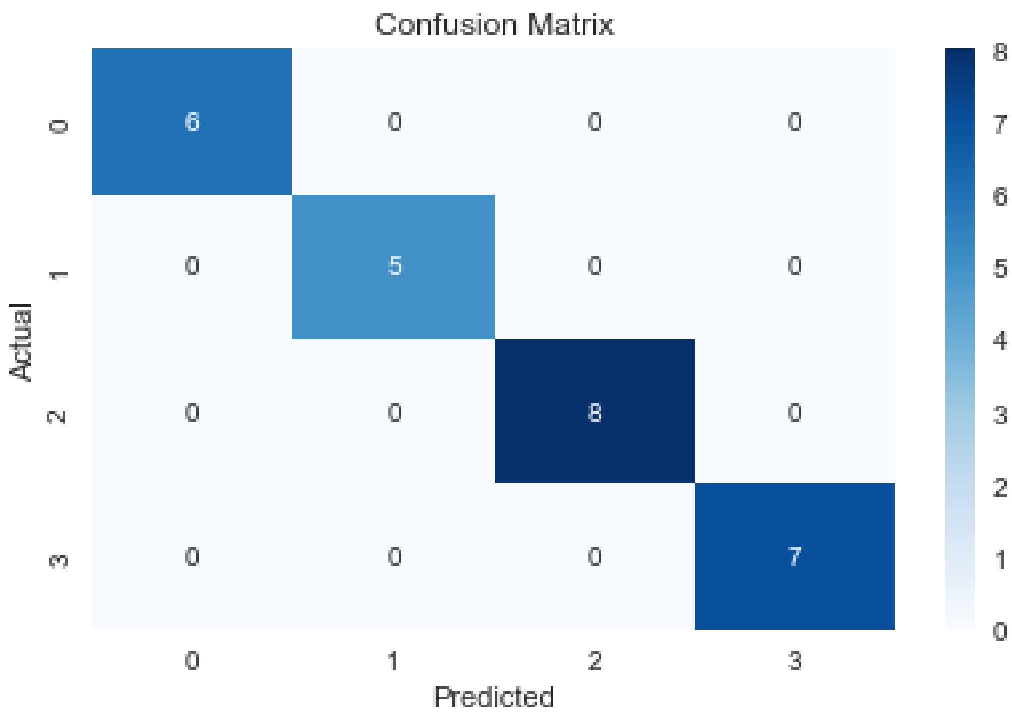
```
Class distribution before resampling:  Gender
3    33
2    31
1    22
0    18
Name: count, dtype: int64
Class distribution after resampling:  Gender
2    33
0    33
1    33
3    33
Name: count, dtype: int64
```

[42]:
```python
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_train_res = scaler.fit_transform(X_train_res)
X_test = scaler.transform(X_test)
```

[44]:
```python
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report, confusion_matrix,␣
 ↪accuracy_score
from sklearn.model_selection import GridSearchCV
knn = KNeighborsClassifier()
param_grid = {
    'n_neighbors': range(1, 21),
    'weights': ['uniform', 'distance'],
    'metric': ['euclidean', 'manhattan', 'minkowski']}

grid_search = GridSearchCV(knn, param_grid, cv=5, scoring='accuracy', n_jobs=-1)
grid_search.fit(X_train_res, y_train_res)
print("Best Parameters:", grid_search.best_params_)
print("Best Cross-Validation Accuracy:", grid_search.best_score_)
best_knn = grid_search.best_estimator_
best_knn.fit(X_train_res, y_train_res)
```

```python
y_pred = best_knn.predict(X_test)

print("\nConfusion Matrix:")
print(confusion_matrix(y_test, y_pred))

print("\nClassification Report:")
print(classification_report(y_test, y_pred))

print("\nAccuracy Score:")
print(accuracy_score(y_test, y_pred))
sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt='d', cmap='Blues')
plt.title("Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```

```
Best Parameters: {'metric': 'euclidean', 'n_neighbors': 1, 'weights': 'uniform'}
Best Cross-Validation Accuracy: 1.0

Confusion Matrix:
[[6 0 0 0]
 [0 5 0 0]
 [0 0 8 0]
 [0 0 0 7]]

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         6
           1       1.00      1.00      1.00         5
           2       1.00      1.00      1.00         8
           3       1.00      1.00      1.00         7

    accuracy                           1.00        26
   macro avg       1.00      1.00      1.00        26
weighted avg       1.00      1.00      1.00        26


Accuracy Score:
1.0
```


Confusion Matrix
Actual
Predicted
0
1
2
3
6
5
8
7