Saloni Sivakumar

+ Code + Text

```
import numpy as np
import pandas as pd
```

```
df=pd.read_csv('/content/Estimates of number of homicides.csv')
df.head()
```

	IndicatorCode	Indicator	ValueType	ParentLocationCode	ParentLocation	Location type	SpatialDimValueCo
0	VIOLENCE_HOMICIDENUM	Estimates of number of homicides	numeric	SEAR	South-East Asia	Country	II
1	VIOLENCE_HOMICIDENUM	Estimates of number of homicides	numeric	SEAR	South-East Asia	Country	II
2	VIOLENCE_HOMICIDENUM	Estimates of number of homicides	numeric	SEAR	South-East Asia	Country	II
3	VIOLENCE_HOMICIDENUM	Estimates of number of homicides	numeric	SEAR	South-East Asia	Country	II
4	VIOLENCE_HOMICIDENUM	Estimates of number of homicides	numeric	SEAR	South-East Asia	Country	II

```
df=df.drop(['IndicatorCode','Indicator','ValueType','ParentLocationCode','ParentLocation','Location type','SpatialDimVa
df.head()
```

	Period	Dim1	FactValueNumeric	FactValueNumericLow	FactValueNumericHigh
0	2019	Female	12509	9259	16578
1	2019	Male	39258	30009	50814
2	2019	Both sexes	51767	39268	67393
3	2018	Female	12477	9443	16283
4	2018	Male	39502	30980	50753

```
df.shape
```

```
(60, 5)
```

```
df.describe()
```

	Period	FactValueNumeric	FactValueNumericLow	FactValueNumericHigh
count	60.000000	60.000000	60.000000	60.000000
mean	2009.500000	35126.000000	28875.300000	42657.466667
std	5.814943	16404.222213	13664.808403	19816.754409
min	2000.000000	12269.000000	9259.000000	15538.000000
25%	2004.750000	14105.750000	11461.500000	17068.500000
50%	2009.500000	39171.000000	32146.500000	46963.500000
75%	2014.250000	52088.250000	42984.000000	62589.250000
max	2019.000000	53957.000000	45198.000000	67393.000000

```
df=df.dropna()
```

```
df.shape
```

```
(60, 5)
```

```
new={'Male':0,'Female':1,'Both sexes':2}
df['Dim1']=df['Dim1'].map(new)
```

```
df.head()
```

	Period	Dim1	FactValueNumeric	FactValueNumericLow	FactValueNumericHigh
0	2019	1	12509	9259	16578
1	2019	0	39258	30009	50814
2	2019	2	51767	39268	67393
3	2018	1	12477	9443	16283
4	2018	0	39502	30980	50753

```
x=df[['Period','FactValueNumeric','FactValueNumericLow','FactValueNumericHigh']] #Separating dependent and independent
y=df['Dim1']
print(x,y)
```

```
    Period  FactValueNumeric  FactValueNumericLow  FactValueNumericHigh
0     2019             12509                 9259                 16578
1     2019             39258                30009                 50814
2     2019             51767                39268                 67393
3     2018             12477                 9443                 16283
4     2018             39502                30980                 50753
5     2018             51979                40423                 67036
6     2017             12433                 9557                 15981
7     2017             39624                31593                 49684
8     2017             52057                41150                 65665
9     2016             12269                 9630                 15538
10    2016             39262                32025                 48265
11    2016             51532                41656                 63803
12    2015             12532                 9997                 15621
13    2015             39782                32976                 48048
14    2015             52314                42973                 63669
15    2014             12875                10400                 15829
16    2014             40435                33891                 48718
17    2014             53309                44291                 64547
18    2013             13027                10672                 15795
19    2013             40444                34051                 48779
20    2013             53470                44724                 64574
21    2012             13295                10938                 15950
22    2012             40610                34179                 48909
```

```
23    2012              53904               45117                64859
24    2011              13583               11101                16358
25    2011              40374               34097                48148
26    2011              53957               45198                64506
27    2010              13745               11278                16503
28    2010              39719               33549                47331
29    2010              53464               44827                63834
30    2009              13924               11499                16691
31    2009              39084               33150                46443
32    2009              53008               44648                63134
33    2008              14105               11640                16904
34    2008              38660               32809                45803
35    2008              52765               44449                62707
36    2007              14111               11546                16955
37    2007              38296               32465                45220
38    2007              52407               44011                62176
39    2006              14052               11473                16974
40    2006              38130               32221                45078
41    2006              52182               43694                62052
42    2005              14008               11407                16956
43    2005              38175               31987                45272
44    2005              52183               43394                62228
45    2004              14106               11538                17100
46    2004              38311               31857                45451
47    2004              52417               43395                62550
48    2003              14020               11360                17123
49    2003              37973               31657                45097
50    2003              51993               43017                62220
51    2002              14247               11427                17580
52    2002              38353               32004                45593
53    2002              52600               43431                63173
54    2001              14370               11340                18105
55    2001              38622               31876                46403
```

```
import matplotlib.pyplot as plt
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score,confusion_matrix
```

```
#Split the dataset into training and testing sets
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=42)
```

```
#Initiate and fit the LDA model
lda=LinearDiscriminantAnalysis()
lda.fit(x_train,y_train)
```

```
▾ LinearDiscriminantAnalysis
LinearDiscriminantAnalysis()
```

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

```
y_pred
```

```
array([1, 2, 1, 1, 0, 1, 1, 1, 1, 1, 0, 2, 0, 1, 0, 2, 2, 1])
```

```
y_test
```

```
0     1
5     2
36    1
45    1
13    0
54    1
33    1
48    1
12    1
```

```
     57    1
     46    0
     50    2
     31    0
     3     1
     52    0
     17    2
     8     2
     6     1
     Name: Dim1, dtype: int64
```

```
accuracy=accuracy_score(y_test,y_pred)
accuracy
```

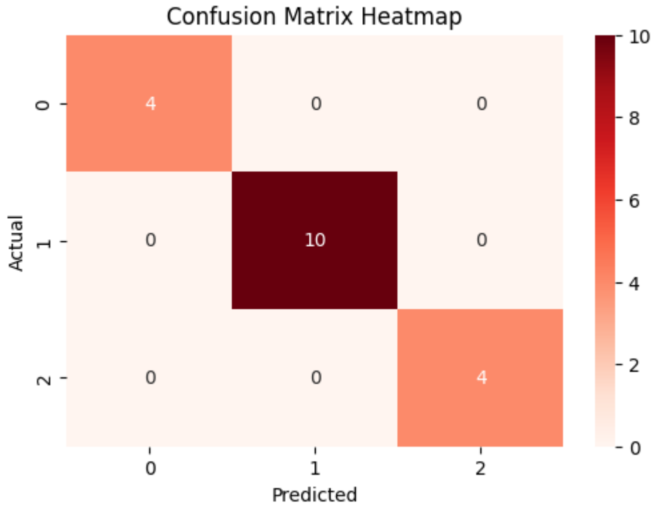
```
     1.0
```

```
confusion=confusion_matrix(y_test,y_pred)
```

```
confusion_matrix(y_test,y_pred)
```

```
     array([[ 4,  0,  0],
            [ 0, 10,  0],
            [ 0,  0,  4]])
```

```
import seaborn as sns
plt.figure(figsize=(6,4))
sns.heatmap(confusion,annot=True,fmt='d',cmap='Reds')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix Heatmap')
plt.show()
```



```
Conclusion:
```

Conclusion: