KNN_drug_classification

February 28, 2025

[ ]:

[289]:
```
import pandas as pd
import numpy  as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, r2_score, mean_squared_error,␣
↪mean_absolute_error
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import GridSearchCV
```

[290]:
```
url = r"D:\Supervised Machine Learning lab (SMLL)\6\Practice dataset 2␣
↪KNNClassifier drug_classification.csv"
df = pd.read_csv(url)
```

[291]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 6 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Age          200 non-null    int64
 1   Sex          200 non-null    object
 2   BP           200 non-null    object
 3   Cholesterol  200 non-null    object
 4   Na_to_K      200 non-null    float64
 5   Drug_Type    200 non-null    object
dtypes: float64(1), int64(1), object(4)
memory usage: 9.5+ KB
```

[292]:
```
df.head()
```

[292]:

	Age	Sex	BP	Cholesterol	Na_to_K	Drug_Type
0	23	F	HIGH	HIGH	25.355	DrugY

```
1   47    M      LOW         HIGH   13.093     drugC
2   47    M      LOW         HIGH   10.114     drugC
3   28    F   NORMAL         HIGH    7.798     drugX
4   61    F      LOW         HIGH   18.043     DrugY
```

[293]:
```python
df.describe(include='all')
```

[293]:

	Age	Sex	BP	Cholesterol	Na_to_K	Drug_Type
count	200.000000	200	200	200	200.000000	200
unique	NaN	2	3	2	NaN	5
top	NaN	M	HIGH	HIGH	NaN	DrugY
freq	NaN	104	77	103	NaN	91
mean	44.315000	NaN	NaN	NaN	16.084485	NaN
std	16.544315	NaN	NaN	NaN	7.223956	NaN
min	15.000000	NaN	NaN	NaN	6.269000	NaN
25%	31.000000	NaN	NaN	NaN	10.445500	NaN
50%	45.000000	NaN	NaN	NaN	13.936500	NaN
75%	58.000000	NaN	NaN	NaN	19.380000	NaN
max	74.000000	NaN	NaN	NaN	38.247000	NaN

[294]:
```python
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
```

[295]:
```python
# Encode categorical columns
df['Sex'] = le.fit_transform(df['Sex'])
df['BP'] = le.fit_transform(df['BP'])
df['Cholesterol'] = le.fit_transform(df['Cholesterol'])
df['Drug_Type'] = le.fit_transform(df['Drug_Type'])

# Display the updated dataframe
df.head()
```

[295]:

	Age	Sex	BP	Cholesterol	Na_to_K	Drug_Type
0	23	0	0	0	25.355	0
1	47	1	1	0	13.093	3
2	47	1	1	0	10.114	3
3	28	0	2	0	7.798	4
4	61	0	1	0	18.043	0

[296]:
```python
X = df.drop('Drug_Type', axis=1)
y = df['Drug_Type']
```

[297]:
```python
x = pd.get_dummies(X, drop_first=True)
```

[298]:
```python
sc = StandardScaler()
x = sc.fit_transform(x)
```

[299]:
```python
x
```

```
[299]: array([[-1.29159102, -1.040833  , -1.11016894, -0.97043679,  1.28652212],
       [ 0.16269866,  0.96076892,  0.10979693, -0.97043679, -0.4151454 ],
       [ 0.16269866,  0.96076892,  0.10979693, -0.97043679, -0.82855818],
       [-0.988614  , -1.040833  ,  1.32976279, -0.97043679, -1.14996267],
       [ 1.0110343 , -1.040833  ,  0.10979693, -0.97043679,  0.27179427],
       [-1.35218642, -1.040833  ,  1.32976279, -0.97043679, -1.03769314],
       [ 0.28388946, -1.040833  ,  1.32976279, -0.97043679,  0.02643885],
       [-0.20087376,  0.96076892,  0.10979693, -0.97043679, -0.70046821],
       [ 0.9504389 ,  0.96076892,  1.32976279, -0.97043679, -0.12676951],
       [-0.07968296,  0.96076892,  0.10979693,  1.03046381,  0.45567206],
       [ 0.16269866, -1.040833  ,  0.10979693, -0.97043679, -0.59916196],
       [-0.62504158, -1.040833  , -1.11016894,  1.03046381,  0.43221897],
       [-0.07968296,  0.96076892,  0.10979693, -0.97043679, -0.09832049],
       [ 1.79877454, -1.040833  ,  0.10979693, -0.97043679,  0.674105  ],
       [ 0.34448487, -1.040833  ,  1.32976279, -0.97043679, -0.46926791],
       [-1.71575884, -1.040833  , -1.11016894,  1.03046381, -0.0788919 ],
       [ 1.49579753,  0.96076892,  0.10979693,  1.03046381, -0.64245998],
       [-0.07968296,  0.96076892, -1.11016894, -0.97043679, -0.29316156],
       [-1.29159102,  0.96076892,  0.10979693, -0.97043679, -1.21935052],
       [-0.74623239, -1.040833  , -1.11016894,  1.03046381,  1.37242427],
       [ 0.76865269,  0.96076892,  0.10979693,  1.03046381,  0.42236589],
       [ 1.13222511,  0.96076892,  1.32976279, -0.97043679,  1.36451406],
       [ 0.16269866,  0.96076892,  0.10979693,  1.03046381,  2.00995979],
       [ 0.22329406, -1.040833  ,  0.10979693, -0.97043679, -0.14550423],
       [-0.68563699, -1.040833  ,  0.10979693, -0.97043679,  2.41490725],
       [-0.988614  , -1.040833  , -1.11016894,  1.03046381,  0.37809645],
       [-0.80682779,  0.96076892, -1.11016894, -0.97043679,  1.9819271 ],
       [ 0.28388946, -1.040833  ,  1.32976279,  1.03046381, -0.93028076],
       [-0.32206457, -1.040833  ,  0.10979693,  1.03046381,  0.91765633],
       [ 0.04150785,  0.96076892,  0.10979693, -0.97043679,  0.25902691],
       [-1.59456803, -1.040833  ,  1.32976279,  1.03046381, -1.01784822],
       [ 1.79877454,  0.96076892, -1.11016894, -0.97043679, -0.90446848],
       [ 0.28388946,  0.96076892,  0.10979693,  1.03046381, -0.70366006],
       [ 1.25341591, -1.040833  , -1.11016894,  1.03046381,  2.19147839],
       [ 0.52627108,  0.96076892,  1.32976279, -0.97043679, -0.27081868],
       [ 0.10210325,  0.96076892,  1.32976279,  1.03046381, -1.2211546 ],
       [-0.74623239,  0.96076892, -1.11016894,  1.03046381, -0.92139911],
       [-0.32206457,  0.96076892,  0.10979693,  1.03046381, -0.29787994],
       [-0.32206457, -1.040833  ,  1.32976279,  1.03046381, -0.88476233],
       [-1.77635424,  0.96076892,  1.32976279, -0.97043679, -0.97149714],
       [ 1.73817914, -1.040833  ,  1.32976279, -0.97043679,  0.43527203],
       [ 0.82924809, -1.040833  , -1.11016894,  1.03046381, -0.25610845],
       [ 0.34448487,  0.96076892,  1.32976279,  1.03046381, -0.04086736],
       [-1.29159102,  0.96076892,  1.32976279, -0.97043679, -0.53074555],
       [ 0.34448487, -1.040833  ,  1.32976279,  1.03046381, -0.5258884 ],
       [ 1.31401132, -1.040833  ,  1.32976279,  1.03046381, -1.10708099],
       [-0.44325537, -1.040833  , -1.11016894, -0.97043679, -0.41542295],
```

```
[ 1.43520212,  0.96076892,  0.10979693, -0.97043679, -0.80399488],
[-1.29159102,  0.96076892,  1.32976279, -0.97043679,  2.16511101],
[-0.988614  , -1.040833  ,  0.10979693, -0.97043679,  0.51506806],
[ 0.82924809, -1.040833  , -1.11016894, -0.97043679,  0.46233329],
[ 1.37460672,  0.96076892,  1.32976279,  1.03046381, -0.71975804],
[ 1.0716297 ,  0.96076892,  0.10979693,  1.03046381,  1.54020408],
[-1.23099561, -1.040833  , -1.11016894,  1.03046381,  0.32924741],
[ 1.43520212, -1.040833  , -1.11016894,  1.03046381, -0.81815   ],
[-1.10980481, -1.040833  ,  0.10979693, -0.97043679, -0.26707173],
[ 1.25341591,  0.96076892, -1.11016894,  1.03046381, -0.65841918],
[-0.26146916,  0.96076892, -1.11016894, -0.97043679,  1.62943685],
[ 0.9504389 ,  0.96076892,  1.32976279,  1.03046381, -0.83175002],
[-0.62504158,  0.96076892, -1.11016894, -0.97043679,  0.36338623],
[-0.38265997, -1.040833  ,  0.10979693,  1.03046381,  1.91378824],
[-1.23099561,  0.96076892, -1.11016894,  1.03046381, -0.91723584],
[ 1.37460672,  0.96076892,  0.10979693,  1.03046381,  0.63954985],
[ 0.04150785,  0.96076892,  0.10979693,  1.03046381, -1.07058298],
[ 0.9504389 , -1.040833  , -1.11016894, -0.97043679, -0.3860025 ],
[ 1.43520212, -1.040833  ,  1.32976279,  1.03046381,  1.52174691],
[-0.9280186 ,  0.96076892, -1.11016894, -0.97043679, -0.44803523],
[-1.65516344,  0.96076892,  1.32976279,  1.03046381, -0.72891723],
[ 0.58686648,  0.96076892,  1.32976279, -0.97043679,  1.18979546],
[-1.59456803, -1.040833  , -1.11016894,  1.03046381,  1.13678315],
[ 1.55639293,  0.96076892, -1.11016894, -0.97043679, -0.29385544],
[-0.988614  , -1.040833  ,  1.32976279, -0.97043679,  0.4982762 ],
[-1.23099561, -1.040833  ,  1.32976279, -0.97043679, -0.76041931],
[-0.20087376, -1.040833  ,  1.32976279,  1.03046381,  0.94652168],
[-0.80682779,  0.96076892, -1.11016894,  1.03046381,  0.13662675],
[-1.10980481,  0.96076892,  0.10979693,  1.03046381,  0.6695254 ],
[-0.50385078, -1.040833  , -1.11016894, -0.97043679, -0.67812533],
[-1.10980481, -1.040833  , -1.11016894,  1.03046381,  0.42694549],
[-1.53397263, -1.040833  , -1.11016894, -0.97043679, -0.38461474],
[-0.74623239, -1.040833  ,  0.10979693,  1.03046381, -0.72780703],
[ 0.9504389 ,  0.96076892, -1.11016894, -0.97043679, -0.29843504],
[ 1.19282051,  0.96076892,  1.32976279, -0.97043679, -1.15509738],
[-0.74623239, -1.040833  ,  0.10979693, -0.97043679, -0.884346  ],
[-0.38265997, -1.040833  , -1.11016894,  1.03046381, -0.66036204],
[ 0.16269866, -1.040833  ,  0.10979693, -0.97043679, -0.83508063],
[ 0.88984349,  0.96076892, -1.11016894, -0.97043679, -0.29829626],
[ 0.40508027, -1.040833  ,  1.32976279, -0.97043679, -0.34520245],
[ 1.49579753,  0.96076892,  0.10979693, -0.97043679, -0.08416537],
[-0.44325537, -1.040833  , -1.11016894,  1.03046381,  0.97233395],
[ 0.34448487, -1.040833  ,  1.32976279,  1.03046381,  0.1563329 ],
[ 1.0716297 ,  0.96076892,  1.32976279, -0.97043679,  0.0707083 ],
[-0.20087376,  0.96076892, -1.11016894,  1.03046381, -0.12885115],
[-0.9280186 , -1.040833  , -1.11016894, -0.97043679,  1.85480857],
[-0.14027836, -1.040833  ,  0.10979693,  1.03046381,  1.82996772],
```

```
[ 0.70805729,  0.96076892,  0.10979693, -0.97043679, -0.14841852],
[-0.50385078,  0.96076892,  0.10979693,  1.03046381, -0.64676202],
[ 0.82924809, -1.040833  ,  0.10979693, -0.97043679,  3.07561832],
[ 0.70805729, -1.040833  , -1.11016894, -0.97043679,  1.29207314],
[-1.47337723,  0.96076892, -1.11016894,  1.03046381,  2.71369132],
[-1.77635424, -1.040833  , -1.11016894,  1.03046381,  0.08888791],
[-0.80682779,  0.96076892, -1.11016894,  1.03046381, -0.58472929],
[ 0.04150785, -1.040833  , -1.11016894, -0.97043679, -0.44831279],
[-0.988614  , -1.040833  ,  0.10979693, -0.97043679, -0.41042702],
[ 0.70805729,  0.96076892,  1.32976279, -0.97043679, -0.98787267],
[-1.35218642,  0.96076892, -1.11016894,  1.03046381,  1.69438387],
[-0.44325537,  0.96076892,  0.10979693,  1.03046381, -0.98759512],
[-1.35218642,  0.96076892,  1.32976279, -0.97043679, -0.57334968],
[-0.14027836,  0.96076892,  0.10979693, -0.97043679,  0.54518238],
[ 1.67758373,  0.96076892, -1.11016894,  1.03046381, -0.88920315],
[-1.29159102,  0.96076892,  1.32976279, -0.97043679,  0.10623487],
[ 0.34448487,  0.96076892, -1.11016894, -0.97043679, -1.19270559],
[ 0.16269866, -1.040833  ,  1.32976279,  1.03046381, -1.30469757],
[-0.56444618,  0.96076892,  0.10979693,  1.03046381, -0.95956243],
[ 1.25341591, -1.040833  ,  0.10979693,  1.03046381, -0.32133303],
[-1.47337723, -1.040833  ,  1.32976279,  1.03046381, -0.94415833],
[ 0.40508027,  0.96076892, -1.11016894, -0.97043679,  0.30676574],
[ 1.37460672,  0.96076892,  1.32976279,  1.03046381, -0.91182359],
[-0.26146916, -1.040833  ,  1.32976279, -0.97043679, -0.83008471],
[-0.74623239, -1.040833  , -1.11016894,  1.03046381, -0.8038561 ],
[ 1.0110343 , -1.040833  , -1.11016894, -0.97043679,  1.3031752 ],
[-0.988614  ,  0.96076892,  1.32976279, -0.97043679,  1.52368977],
[-1.77635424,  0.96076892, -1.11016894,  1.03046381,  0.15563902],
[-0.62504158,  0.96076892,  1.32976279, -0.97043679,  0.88421139],
[-0.50385078, -1.040833  ,  1.32976279, -0.97043679,  0.09277363],
[ 0.52627108, -1.040833  , -1.11016894,  1.03046381, -0.49813326],
[-1.53397263, -1.040833  , -1.11016894,  1.03046381,  1.37173039],
[ 1.31401132,  0.96076892, -1.11016894, -0.97043679,  0.0364307 ],
[-0.56444618,  0.96076892,  1.32976279,  1.03046381, -1.14344022],
[ 0.16269866,  0.96076892,  0.10979693,  1.03046381,  2.42267869],
[-0.74623239, -1.040833  ,  1.32976279, -0.97043679, -1.19450967],
[ 1.55639293, -1.040833  ,  1.32976279, -0.97043679,  0.61123961],
[ 0.46567567,  0.96076892,  0.10979693,  1.03046381,  2.33663776],
[ 0.28388946,  0.96076892,  0.10979693,  1.03046381, -0.34506367],
[-1.23099561,  0.96076892,  1.32976279, -0.97043679,  1.34633444],
[-0.14027836, -1.040833  , -1.11016894, -0.97043679,  0.68714991],
[ 1.79877454,  0.96076892,  0.10979693,  1.03046381, -0.57529254],
[ 0.64746188, -1.040833  , -1.11016894, -0.97043679, -0.70879476],
[-0.56444618, -1.040833  , -1.11016894, -0.97043679, -0.44276176],
[ 0.40508027,  0.96076892, -1.11016894,  1.03046381, -0.65800285],
[ 1.49579753, -1.040833  ,  1.32976279, -0.97043679, -0.83535819],
[ 0.28388946,  0.96076892, -1.11016894,  1.03046381, -1.36215071],
```

```
[ 1.19282051, -1.040833  ,  0.10979693,  1.03046381,  1.34008953],
[ 0.9504389 ,  0.96076892, -1.11016894,  1.03046381, -1.03575028],
[ 1.79877454,  0.96076892, -1.11016894,  1.03046381, -0.08999395],
[-0.32206457,  0.96076892, -1.11016894, -0.97043679, -0.89100724],
[ 1.0110343 ,  0.96076892,  1.32976279, -0.97043679, -0.92167666],
[-0.44325537, -1.040833  ,  0.10979693,  1.03046381, -0.56599457],
[-1.10980481, -1.040833  , -1.11016894,  1.03046381, -0.52422309],
[ 1.0110343 , -1.040833  ,  0.10979693,  1.03046381, -1.21352194],
[-1.35218642,  0.96076892,  0.10979693, -0.97043679, -1.10097486],
[ 0.28388946,  0.96076892, -1.11016894,  1.03046381, -1.024787  ],
[ 1.43520212,  0.96076892, -1.11016894, -0.97043679, -0.70435393],
[ 0.64746188,  0.96076892,  1.32976279,  1.03046381, -1.22448522],
[ 1.67758373, -1.040833  ,  0.10979693,  1.03046381, -0.20018185],
[-0.44325537,  0.96076892,  0.10979693,  1.03046381,  0.08874914],
[ 0.28388946,  0.96076892,  0.10979693, -0.97043679, -0.76985606],
[-0.80682779,  0.96076892, -1.11016894,  1.03046381, -0.67410083],
[ 0.52627108,  0.96076892,  0.10979693, -0.97043679,  0.95457067],
[ 0.88984349, -1.040833  ,  0.10979693, -0.97043679, -0.7827622 ],
[-0.62504158, -1.040833  ,  0.10979693,  1.03046381, -0.43873726],
[-0.8674232 , -1.040833  ,  1.32976279, -0.97043679, -0.78290097],
[ 0.76865269, -1.040833  , -1.11016894,  1.03046381, -0.85201127],
[-0.07968296,  0.96076892,  1.32976279,  1.03046381, -0.44761891],
[-1.41278182, -1.040833  , -1.11016894,  1.03046381,  1.74129005],
[-1.71575884,  0.96076892, -1.11016894,  1.03046381,  0.40557404],
[-0.38265997,  0.96076892,  0.10979693, -0.97043679,  0.30676574],
[ 0.82924809, -1.040833  ,  0.10979693, -0.97043679,  1.46554276],
[ 0.76865269, -1.040833  ,  1.32976279, -0.97043679, -0.2593003 ],
[ 0.40508027, -1.040833  ,  0.10979693,  1.03046381,  0.96012169],
[-1.47337723, -1.040833  , -1.11016894, -0.97043679, -0.66924368],
[-0.988614  , -1.040833  ,  1.32976279, -0.97043679, -0.44484339],
[ 0.04150785,  0.96076892,  0.10979693,  1.03046381, -0.84201942],
[-0.32206457, -1.040833  ,  1.32976279,  1.03046381,  0.15827576],
[-0.20087376, -1.040833  ,  0.10979693,  1.03046381,  0.36838215],
[-0.14027836,  0.96076892, -1.11016894,  1.03046381, -0.46052505],
[ 1.73817914, -1.040833  , -1.11016894, -0.97043679,  0.31412086],
[ 0.22329406,  0.96076892, -1.11016894,  1.03046381, -0.78248465],
[-1.17040021,  0.96076892,  1.32976279, -0.97043679,  0.40612914],
[-0.32206457,  0.96076892,  1.32976279, -0.97043679, -0.01602651],
[ 1.37460672, -1.040833  ,  1.32976279, -0.97043679, -0.02685101],
[-1.35218642, -1.040833  , -1.11016894,  1.03046381,  0.93444819],
[ 0.88984349, -1.040833  ,  1.32976279, -0.97043679, -0.30537382],
[-1.47337723, -1.040833  ,  0.10979693,  1.03046381, -0.61040279],
[-0.50385078, -1.040833  , -1.11016894,  1.03046381, -0.08250007],
[-1.59456803, -1.040833  , -1.11016894, -0.97043679,  2.92865486],
[ 0.76865269, -1.040833  ,  1.32976279,  1.03046381,  1.36118344],
[ 1.55639293,  0.96076892, -1.11016894, -0.97043679, -0.86533373],
[ 0.16269866,  0.96076892, -1.11016894, -0.97043679, -0.788452  ],
```

```
       [ 1.25341591,  0.96076892, -1.11016894,  1.03046381,  2.62459732],
       [ 1.19282051,  0.96076892, -1.11016894,  1.03046381,  0.67271724],
       [ 0.82924809,  0.96076892, -1.11016894, -0.97043679,  0.40335363],
       [-1.29159102,  0.96076892, -1.11016894, -0.97043679, -1.12040345],
       [ 1.67758373,  0.96076892,  0.10979693, -0.97043679,  0.031296  ],
       [ 1.67758373,  0.96076892,  0.10979693, -0.97043679, -1.29276286],
       [ 0.10210325, -1.040833  , -1.11016894, -0.97043679,  2.58143808],
       [ 0.70805729, -1.040833  ,  0.10979693, -0.97043679, -0.6269171 ],
       [-1.71575884,  0.96076892,  0.10979693, -0.97043679, -0.56599457],
       [ 0.46567567,  0.96076892,  1.32976279, -0.97043679, -0.85908883],
       [-1.29159102,  0.96076892,  1.32976279,  1.03046381, -0.28650033],
       [-0.26146916, -1.040833  ,  0.10979693,  1.03046381, -0.6571702 ]])
```

[300]:
```python
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.3,␣
 ↪random_state=42, shuffle=True)
```

[301]:
```python
params = {
    'n_neighbors':np.array(range(1, 50)),
    'weights' : ['uniform', 'distance'],
    'metric':['minkowski','manhattan','euclidean']
}
```

[302]:
```python
from sklearn.neighbors import KNeighborsClassifier
dia_reg = GridSearchCV(KNeighborsClassifier(), params, cv = 10)
```

[303]:
```python
dia_reg.fit(X_train, y_train)
```

[303]:
```
GridSearchCV(cv=10, estimator=KNeighborsClassifier(),
             param_grid={'metric': ['minkowski', 'manhattan', 'euclidean'],
                         'n_neighbors': array([ 1,  2,  3,  4,  5,  6,  7,  8,
9, 10, 11, 12, 13, 14, 15, 16, 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]),
                         'weights': ['uniform', 'distance']})
```

[304]:
```python
dia_reg.best_score_
```

[304]:
```
0.8928571428571429
```

[305]:
```python
dia_reg.best_params_
```

[305]:
```
{'metric': 'manhattan', 'n_neighbors': 21, 'weights': 'distance'}
```

[306]:
```python
regressor = KNeighborsClassifier(metric = 'manhattan', n_neighbors= 21,␣
 ↪weights='distance')
regressor.fit(X_train, y_train)
```

[306]:
```
KNeighborsClassifier(metric='manhattan', n_neighbors=21, weights='distance')
```

[307]:
```python
y_pred = regressor.predict(X_test)
```

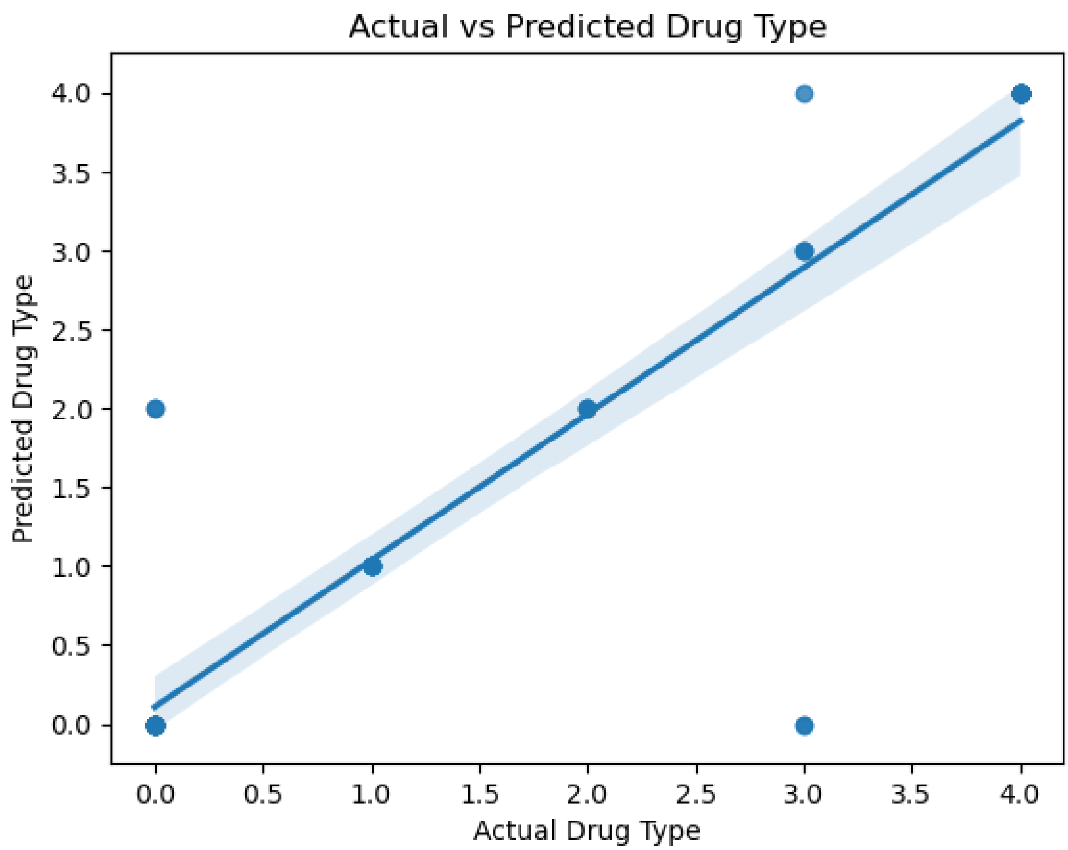
[308]:
```python
from sklearn.metrics import accuracy_score
# Calculate and display accuracy score
print("Accuracy score : {:.4f}".format(accuracy_score(y_test, y_pred)))
```

```
Accuracy score : 0.9167
```

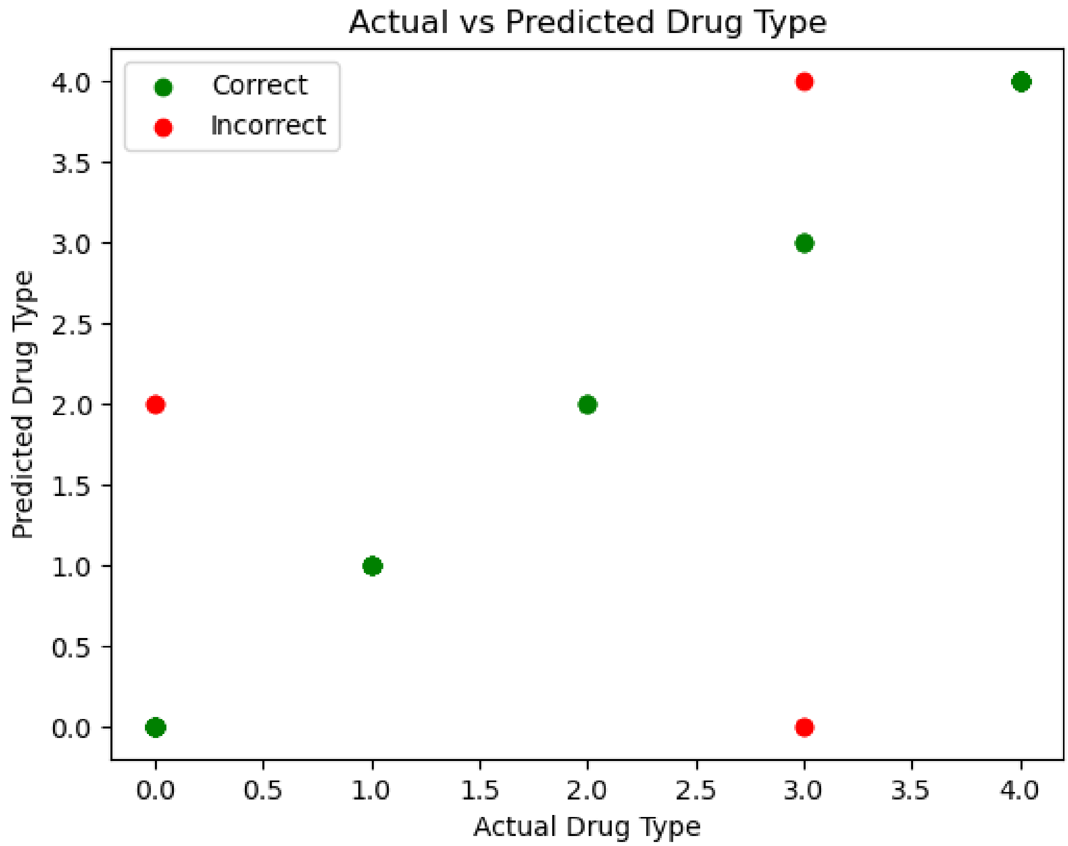
[309]:
```python
from sklearn.metrics import confusion_matrix, classification_report

# Evaluate the model
print("Confusion 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))
```

```
Confusion Matrix:
[[24  0  2  0  0]
 [ 0  7  0  0  0]
 [ 0  0  3  0  0]
 [ 2  0  0  3  1]
 [ 0  0  0  0 18]]

Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.92      0.92        26
           1       1.00      1.00      1.00         7
           2       0.60      1.00      0.75         3
           3       1.00      0.50      0.67         6
           4       0.95      1.00      0.97        18

    accuracy                           0.92        60
   macro avg       0.89      0.88      0.86        60
weighted avg       0.93      0.92      0.91        60


Accuracy Score:
0.9166666666666666
```

[310]:
```python
# Plot the regplot for the predictions
sns.regplot(x=y_test, y=y_pred)
plt.xlabel('Actual Drug Type')
plt.ylabel('Predicted Drug Type')
plt.title('Actual vs Predicted Drug Type')
plt.show()
```



[311]:
```
# Create a boolean array indicating correct predictions
correct = y_test == y_pred

# Plot the correct predictions in green
plt.scatter(y_test[correct], y_pred[correct], color='green', label='Correct')

# Plot the incorrect predictions in red
plt.scatter(y_test[~correct], y_pred[~correct], color='red', label='Incorrect')

# Add labels and title
plt.xlabel('Actual Drug Type')
plt.ylabel('Predicted Drug Type')
plt.title('Actual vs Predicted Drug Type')

# Add legend
plt.legend()

# Show plot
plt.show()
```


Actual vs Predicted Drug Type
Correct
Incorrect
Predicted Drug Type
Actual Drug Type
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0