knn-star-classify

[4]:

```
#importing libraries and creating dataframe of the dataset.

import numpy as np
import pandas as pd
df=pd.read_csv('/content/star_classification.csv')
df
```

[4]:

```
             obj_ID       alpha      delta         u         g         r  \
0      1.237661e+18  135.689107  32.494632  23.87882  22.27530  20.39501
1      1.237665e+18  144.826101  31.274185  24.77759  22.83188  22.58444
2      1.237661e+18  142.188790  35.582444  25.26307  22.66389  20.60976
3      1.237663e+18  338.741038  -0.402828  22.13682  23.77656  21.61162
4      1.237680e+18  345.282593  21.183866  19.43718  17.58028  16.49747
…             …         …        …       …       …       …
99995  1.237679e+18   39.620709  -2.594074  22.16759  22.97586  21.90404
99996  1.237679e+18   29.493819  19.798874  22.69118  22.38628  20.45003
99997  1.237668e+18  224.587407  15.700707  21.16916  19.26997  18.20428
99998  1.237661e+18  212.268621  46.660365  25.35039  21.63757  19.91386
99999  1.237661e+18  196.896053  49.464643  22.62171  21.79745  20.60115

              i         z  run_ID  rerun_ID  cam_col  field_ID   spec_obj_ID  \
0      19.16573  18.79371    3606       301        2        79  6.543777e+18
1      21.16812  21.61427    4518       301        5       119  1.176014e+19
2      19.34857  18.94827    3606       301        2       120  5.152200e+18
3      20.50454  19.25010    4192       301        3       214  1.030107e+19
4      15.97711  15.54461    8102       301        3       137  6.891865e+18
…         …       …     …       …      …       …           …
99995  21.30548  20.73569    7778       301        2       581  1.055431e+19
99996  19.75759  19.41526    7917       301        1       289  8.586351e+18
99997  17.69034  17.35221    5314       301        4       308  3.112008e+18
99998  19.07254  18.62482    3650       301        4       131  7.601080e+18
99999  20.00959  19.28075    3650       301        4        60  8.343152e+18

        class  redshift  plate    MJD  fiber_ID
0      GALAXY  0.634794   5812  56354       171
1      GALAXY  0.779136  10445  58158       427
2      GALAXY  0.644195   4576  55592       299
```

```
3      GALAXY  0.932346   9149  58039       775
4      GALAXY  0.116123   6121  56187       842
…        …        …    …    …     …
99995  GALAXY  0.000000   9374  57749       438
99996  GALAXY  0.404895   7626  56934       866
99997  GALAXY  0.143366   2764  54535        74
99998  GALAXY  0.455040   6751  56368       470
99999  GALAXY  0.542944   7410  57104       851

[100000 rows x 18 columns]
```

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

[5]:
```
         obj_ID       alpha      delta         u         g         r  \
0  1.237661e+18  135.689107  32.494632  23.87882  22.27530  20.39501
1  1.237665e+18  144.826101  31.274185  24.77759  22.83188  22.58444
2  1.237661e+18  142.188790  35.582444  25.26307  22.66389  20.60976
3  1.237663e+18  338.741038  -0.402828  22.13682  23.77656  21.61162
4  1.237680e+18  345.282593  21.183866  19.43718  17.58028  16.49747

          i         z  run_ID  rerun_ID  cam_col  field_ID   spec_obj_ID  \
0  19.16573  18.79371    3606       301        2        79  6.543777e+18
1  21.16812  21.61427    4518       301        5       119  1.176014e+19
2  19.34857  18.94827    3606       301        2       120  5.152200e+18
3  20.50454  19.25010    4192       301        3       214  1.030107e+19
4  15.97711  15.54461    8102       301        3       137  6.891865e+18

    class  redshift  plate    MJD  fiber_ID
0  GALAXY  0.634794   5812  56354       171
1  GALAXY  0.779136  10445  58158       427
2  GALAXY  0.644195   4576  55592       299
3  GALAXY  0.932346   9149  58039       775
4  GALAXY  0.116123   6121  56187       842
```

[6]:
```
df.tail()
```

[6]:
```
             obj_ID       alpha      delta         u         g         r  \
99995  1.237679e+18   39.620709  -2.594074  22.16759  22.97586  21.90404
99996  1.237679e+18   29.493819  19.798874  22.69118  22.38628  20.45003
99997  1.237668e+18  224.587407  15.700707  21.16916  19.26997  18.20428
99998  1.237661e+18  212.268621  46.660365  25.35039  21.63757  19.91386
99999  1.237661e+18  196.896053  49.464643  22.62171  21.79745  20.60115

              i         z  run_ID  rerun_ID  cam_col  field_ID   spec_obj_ID  \
99995  21.30548  20.73569    7778       301        2       581  1.055431e+19
99996  19.75759  19.41526    7917       301        1       289  8.586351e+18
99997  17.69034  17.35221    5314       301        4       308  3.112008e+18
```

```
99998  19.07254  18.62482    3650       301         4        131  7.601080e+18
99999  20.00959  19.28075    3650       301         4         60  8.343152e+18

        class  redshift  plate    MJD  fiber_ID
99995  GALAXY  0.000000   9374  57749       438
99996  GALAXY  0.404895   7626  56934       866
99997  GALAXY  0.143366   2764  54535        74
99998  GALAXY  0.455040   6751  56368       470
99999  GALAXY  0.542944   7410  57104       851
```

[7]:
```
df.shape
```

[7]:
```
(100000, 18)
```

[8]:
```
df.columns
```

[8]:
```
Index(['obj_ID', 'alpha', 'delta', 'u', 'g', 'r', 'i', 'z', 'run_ID',
       'rerun_ID', 'cam_col', 'field_ID', 'spec_obj_ID', 'class', 'redshift',
       'plate', 'MJD', 'fiber_ID'],
      dtype='object')
```

[9]:
```
#checking for missing values

df.isna().sum()
```

[9]:
```
obj_ID         0
alpha          0
delta          0
u              0
g              0
r              0
i              0
z              0
run_ID         0
rerun_ID       0
cam_col        0
field_ID       0
spec_obj_ID    0
class          0
redshift       0
plate          0
MJD            0
fiber_ID       0
dtype: int64
```

[10]:
```
df.dtypes
```

```
[10]: obj_ID         float64
      alpha          float64
      delta          float64
      u              float64
      g              float64
      r              float64
      i              float64
      z              float64
      run_ID           int64
      rerun_ID         int64
      cam_col          int64
      field_ID         int64
      spec_obj_ID    float64
      class           object
      redshift       float64
      plate            int64
      MJD              int64
      fiber_ID         int64
      dtype: object
```

```
[11]: df.drop(['obj_ID'],axis=1,inplace=True)
      df
```

```
[11]:             alpha      delta         u         g         r         i  \
      0      135.689107  32.494632  23.87882  22.27530  20.39501  19.16573
      1      144.826101  31.274185  24.77759  22.83188  22.58444  21.16812
      2      142.188790  35.582444  25.26307  22.66389  20.60976  19.34857
      3      338.741038  -0.402828  22.13682  23.77656  21.61162  20.50454
      4      345.282593  21.183866  19.43718  17.58028  16.49747  15.97711
      …             …        …        …        …        …        …
      99995   39.620709  -2.594074  22.16759  22.97586  21.90404  21.30548
      99996   29.493819  19.798874  22.69118  22.38628  20.45003  19.75759
      99997  224.587407  15.700707  21.16916  19.26997  18.20428  17.69034
      99998  212.268621  46.660365  25.35039  21.63757  19.91386  19.07254
      99999  196.896053  49.464643  22.62171  21.79745  20.60115  20.00959

                    z  run_ID  rerun_ID  cam_col  field_ID   spec_obj_ID   class  \
      0      18.79371    3606       301        2        79  6.543777e+18  GALAXY
      1      21.61427    4518       301        5       119  1.176014e+19  GALAXY
      2      18.94827    3606       301        2       120  5.152200e+18  GALAXY
      3      19.25010    4192       301        3       214  1.030107e+19  GALAXY
      4      15.54461    8102       301        3       137  6.891865e+18  GALAXY
      …           …     …       …       …       …        …        …
      99995  20.73569    7778       301        2       581  1.055431e+19  GALAXY
      99996  19.41526    7917       301        1       289  8.586351e+18  GALAXY
      99997  17.35221    5314       301        4       308  3.112008e+18  GALAXY
      99998  18.62482    3650       301        4       131  7.601080e+18  GALAXY
```

```
99999  19.28075   3650      301       4        60  8.343152e+18  GALAXY

       redshift  plate    MJD  fiber_ID
0      0.634794   5812  56354       171
1      0.779136  10445  58158       427
2      0.644195   4576  55592       299
3      0.932346   9149  58039       775
4      0.116123   6121  56187       842
…           …     …     …       …
99995  0.000000   9374  57749       438
99996  0.404895   7626  56934       866
99997  0.143366   2764  54535        74
99998  0.455040   6751  56368       470
99999  0.542944   7410  57104       851

[100000 rows x 17 columns]
```

[12]:

```
#Splitting the dataframe into input and output features

x=df.drop(['class'],axis=1).values
x
```

[12]:

```
array([[1.35689107e+02, 3.24946318e+01, 2.38788200e+01, …,
        5.81200000e+03, 5.63540000e+04, 1.71000000e+02],
       [1.44826101e+02, 3.12741849e+01, 2.47775900e+01, …,
        1.04450000e+04, 5.81580000e+04, 4.27000000e+02],
       [1.42188790e+02, 3.55824442e+01, 2.52630700e+01, …,
        4.57600000e+03, 5.55920000e+04, 2.99000000e+02],
       …,
       [2.24587407e+02, 1.57007074e+01, 2.11691600e+01, …,
        2.76400000e+03, 5.45350000e+04, 7.40000000e+01],
       [2.12268621e+02, 4.66603653e+01, 2.53503900e+01, …,
        6.75100000e+03, 5.63680000e+04, 4.70000000e+02],
       [1.96896053e+02, 4.94646428e+01, 2.26217100e+01, …,
        7.41000000e+03, 5.71040000e+04, 8.51000000e+02]])
```

[13]:

```
y=df['class'].values
y
```

[13]:

```
array(['GALAXY', 'GALAXY', 'GALAXY', …, 'GALAXY', 'GALAXY', 'GALAXY'],
      dtype=object)
```

[14]:

```
#Splitting the features into training and testing datas.

from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.
 ↪30,random_state=42)
```

```
x_train
```

```
[14]: array([[1.30932167e+02, 4.31341083e+00, 2.01307000e+01, …,
                1.18800000e+03, 5.26500000e+04, 1.65000000e+02],
               [2.25429599e+02, 3.31720833e+01, 1.98631500e+01, …,
                2.93500000e+03, 5.46520000e+04, 3.74000000e+02],
               [2.19173525e+02, 5.55117400e+01, 1.98478500e+01, …,
                3.29600000e+03, 5.49090000e+04, 4.75000000e+02],
               …,
               [1.56991726e+02, 3.86124564e+01, 2.39568400e+01, …,
                3.26200000e+03, 5.48840000e+04, 5.41000000e+02],
               [5.58294316e+01, 9.76439658e+00, 1.77922400e+01, …,
                2.67900000e+03, 5.43680000e+04, 2.87000000e+02],
               [1.89902619e+02, 3.37795907e+01, 2.49314200e+01, …,
                3.97100000e+03, 5.53220000e+04, 4.50000000e+02]])
```

```
[15]: x_test
```

```
[15]: array([[1.69568898e+01, 3.64613009e+00, 2.33354200e+01, …,
                4.31200000e+03, 5.55110000e+04, 4.95000000e+02],
               [2.40063240e+02, 6.13413060e+00, 1.78603300e+01, …,
                2.17500000e+03, 5.46120000e+04, 3.48000000e+02],
               [3.08872221e+01, 1.18870964e+00, 1.81891100e+01, …,
                7.33200000e+03, 5.66830000e+04, 9.43000000e+02],
               …,
               [2.09415904e+02, 4.98478278e+01, 2.29654700e+01, …,
                7.43200000e+03, 5.71070000e+04, 8.67000000e+02],
               [2.26833308e+02, 2.61099640e+01, 1.94337400e+01, …,
                2.15400000e+03, 5.45390000e+04, 2.54000000e+02],
               [4.69558448e+01, 8.55015509e-01, 2.26332100e+01, …,
                1.06600000e+03, 5.25890000e+04, 3.78000000e+02]])
```

```
[16]: #Normalization by Standard Scaler

      from sklearn.preprocessing import StandardScaler
      scaler=StandardScaler()
      scaler.fit(x_train)
      x_train=scaler.transform(x_train)
      x_test=scaler.transform(x_test)
      x_train
```

```
[16]: array([[-0.48390605, -1.00945814, -0.86643846, …, -1.33420844,
               -1.62135443, -1.04171225],
              [ 0.49756661,  0.45809196, -0.98519609, …, -0.74322585,
               -0.51517188, -0.27483231],
              [ 0.43258954,  1.59413054, -0.99198732, …, -0.62110523,
               -0.37316943,  0.09576517],
              …,
              [-0.21324531,  0.73475126,  0.83187338, …, -0.6326069 ,
               -0.3869829 ,  0.33793778],
              [-1.26394079, -0.73225915, -1.90441259, …, -0.82982662,
               -0.67209288, -0.59405984],
              [ 0.128575  ,  0.48898554,  1.26446099, …, -0.39276336,
               -0.14497093,  0.00403312]])
```

[17]:
```
x_test
```

[17]:
```
array([[-1.6676802 , -1.04339136,  0.55604319, …, -0.27740843,
        -0.04054111,  0.16915081],
       [ 0.64955506, -0.91686907, -1.87418942, …, -1.00032188,
        -0.53727343, -0.37023364],
       [-1.52299647, -1.16835857, -1.72825358, …,  0.74421002,
         0.60703429,  1.81298913],
       …,
       [ 0.33124459,  1.30610361,  0.39183318, …,  0.77803845,
         0.84131071,  1.5341237 ],
       [ 0.51214587,  0.098962  , -1.17579866, …, -1.00742585,
        -0.57760876, -0.71514615],
       [-1.35610396, -1.18532792,  0.24435267, …, -1.37547912,
        -1.65505929, -0.26015518]])
```

[18]:
```
#Model creation

from sklearn.neighbors import KNeighborsClassifier
knn=KNeighborsClassifier(n_neighbors=7)
knn.fit(x_train,y_train)
y_pred=knn.predict(x_test)
y_pred
```

[18]:
```
array(['GALAXY', 'STAR', 'STAR', …, 'STAR', 'QSO', 'STAR'], dtype=object)
```

[19]:
```
y_test
```

[19]:
```
array(['GALAXY', 'STAR', 'STAR', …, 'STAR', 'QSO', 'STAR'], dtype=object)
```

[20]:
```
y_train
```

[20]:
```
array(['GALAXY', 'STAR', 'STAR', …, 'STAR', 'GALAXY', 'GALAXY'],
      dtype=object)
```

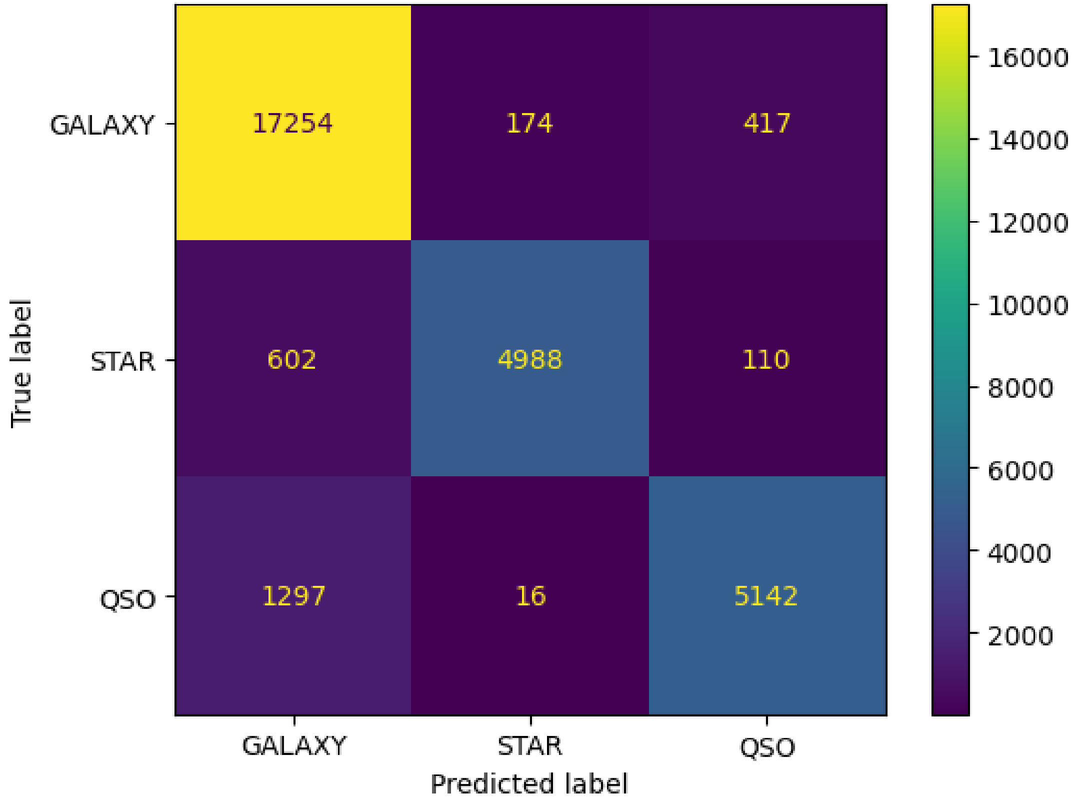
[21]:
```
#Performance Evaluation

from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_test,y_pred)
```

```
cm
```

```
[21]: array([[17254,   174,   417],
             [  602,  4988,   110],
             [ 1297,    16,  5142]])
```

```
[22]: from sklearn.metrics import ConfusionMatrixDisplay
      labels=['GALAXY','STAR','QSO']
      cmd=ConfusionMatrixDisplay(cm,display_labels=labels)
      cmd.plot()
```

```
[22]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at
      0x7b365f8da0b0>
```



```
[23]: from sklearn.metrics import accuracy_score
      score=accuracy_score(y_test,y_pred)
      score
```

```
[23]: 0.9128
```

[24]:

```
from sklearn.metrics import classification_report
report=classification_report(y_test,y_pred)
print(report)
```

```
              precision    recall  f1-score   support

      GALAXY       0.90      0.97      0.93     17845
         QSO       0.96      0.88      0.92      5700
        STAR       0.91      0.80      0.85      6455

    accuracy                           0.91     30000
   macro avg       0.92      0.88      0.90     30000
weighted avg       0.91      0.91      0.91     30000
```