```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
from sklearn.model_selection import train_test_split
```

```
df=pd.read_csv('/content/star_classification.csv')
```

```
df=df.drop(['obj_ID'],axis=1)
```

```
df.head()
```

	alpha	delta	u	g	r	i	z	run_ID	reru
0	135.689107	32.494632	23.87882	22.27530	20.39501	19.16573	18.79371	3606	
1	144.826101	31.274185	24.77759	22.83188	22.58444	21.16812	21.61427	4518	
2	142.188790	35.582444	25.26307	22.66389	20.60976	19.34857	18.94827	3606	
3	338.741038	-0.402828	22.13682	23.77656	21.61162	20.50454	19.25010	4192	
4	345.282593	21.183866	19.43718	17.58028	16.49747	15.97711	15.54461	8102	

```
df.tail()
```

	alpha	delta	u	g	r	i	z	run_ID	rerun_ID	cam_col	field_ID	
99995	39.620709	-2.594074	22.16759	22.97586	21.90404	21.30548	20.73569	7778	301	2	581	1
99996	29.493819	19.798874	22.69118	22.38628	20.45003	19.75759	19.41526	7917	301	1	289	8
99997	224.587407	15.700707	21.16916	19.26997	18.20428	17.69034	17.35221	5314	301	4	308	3
99998	212.268621	46.660365	25.35039	21.63757	19.91386	19.07254	18.62482	3650	301	4	131	7
99999	196.896053	49.464643	22.62171	21.79745	20.60115	20.00959	19.28075	3650	301	4	60	8

```
df.shape
```

```
(100000, 17)
```

```
df.describe
```

```
<bound method NDFrame.describe of             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]>
```

```
df.dtypes
```

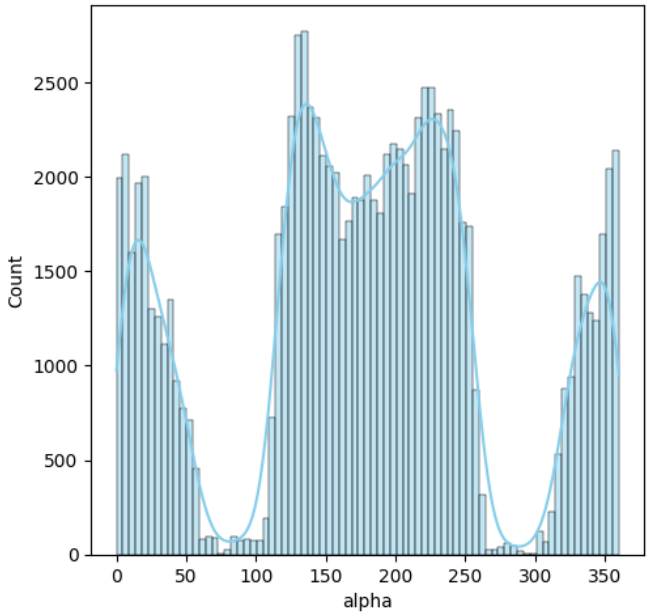
```
alpha          float64
delta          float64
u              float64
g              float64
r              float64
i              float64
z              float64
run_ID           int64
rerun_ID         int64
cam_col          int64
field_ID         int64
spec_obj_ID    float64
class           object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

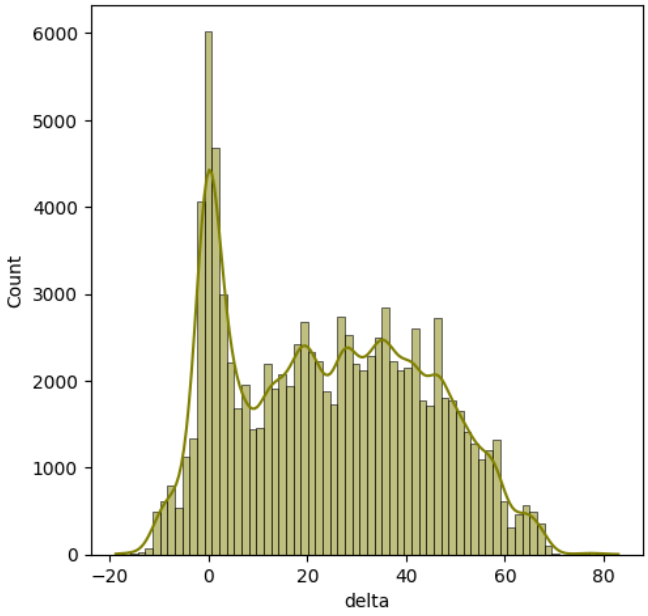
```
df.isna().sum()
```

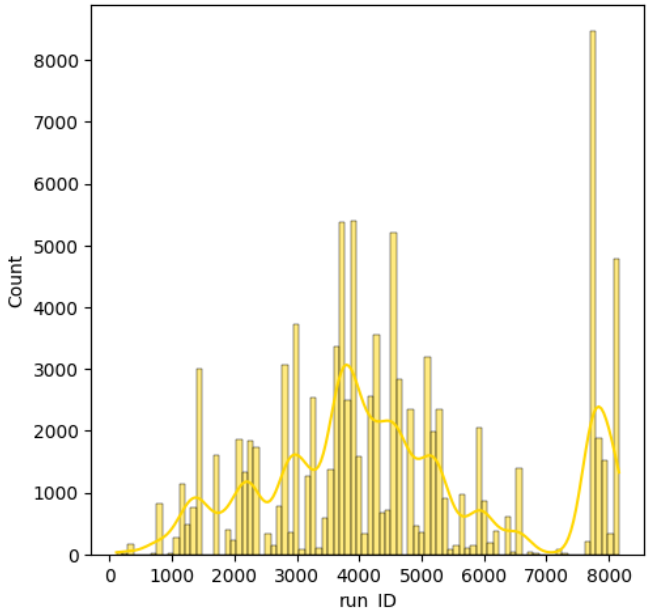
```
alpha          0
delta          0
u              0
g              0
r              0
i              0
z              0
run_ID         0
rerun_ID       0
cam_col        0
field_ID       0
spec_obj_ID    0
class          0
redshift       0
plate          0
MJD            0
fiber_ID       0
dtype: int64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

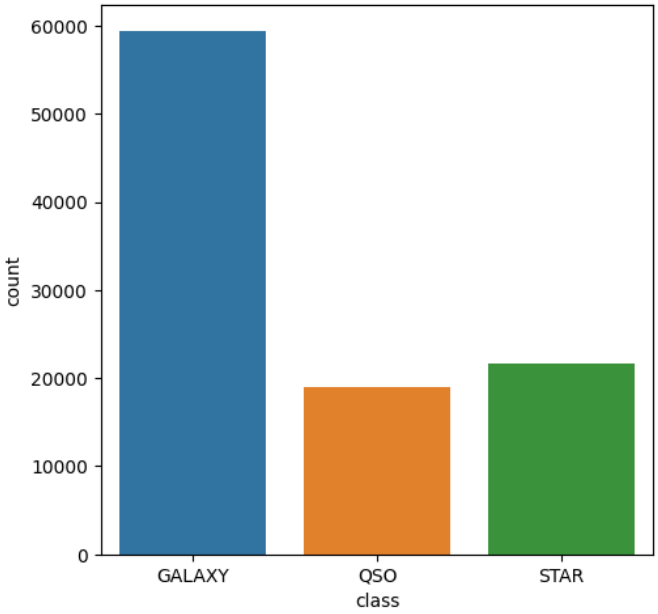
```
fig, axs = plt.subplots(2, 2, figsize=(12, 12))
sns.histplot(data=df["alpha"], kde=True, color="skyblue", ax=axs[0, 0])
sns.histplot(data=df["delta"], kde=True, color="olive", ax=axs[0, 1])
sns.histplot(data=df["run_ID"], kde=True, color="gold", ax=axs[1, 0])
sns.countplot(x = df["class"], ax=axs[1, 1])
```

```
<Axes: xlabel='class', ylabel='count'>
```









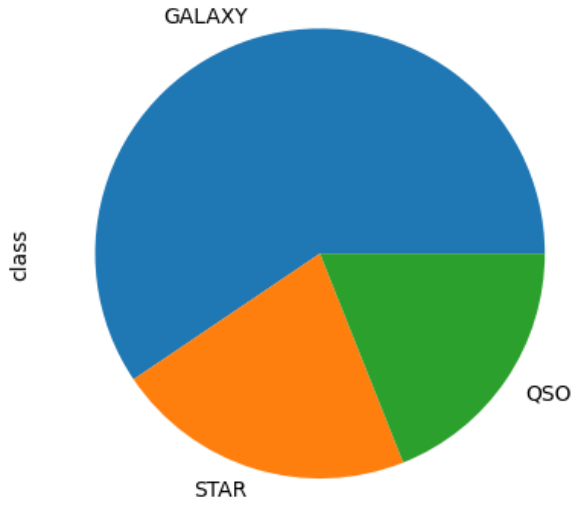

```
x=df.drop(['class'],axis=1)
y=df['class']
```

```
x
```

	alpha	delta	u	g	r	i	z	run_ID	rerun_ID	cam_col	field_ID
0	135.689107	32.494632	23.87882	22.27530	20.39501	19.16573	18.79371	3606	301	2	79
1	144.826101	31.274185	24.77759	22.83188	22.58444	21.16812	21.61427	4518	301	5	119

```
df['class'].value_counts().plot(kind='pie')
```

```
<Axes: ylabel='class'>
```




```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.30,random_state=42)
```

```
y_train
```

```
76513    GALAXY
60406      STAR
27322      STAR
53699    GALAXY
65412    GALAXY
          ...
6265       QSO
54886    GALAXY
76820      STAR
860      GALAXY
15795    GALAXY
Name: class, Length: 70000, dtype: object
```

```
from sklearn.preprocessing import StandardScaler
scalar=StandardScaler()

scalar.fit(x_train)

x_train=scalar.fit_transform(x_train)
x_test=scalar.fit_transform(x_test)


from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier

model1=KNeighborsClassifier(n_neighbors=9,weights='uniform')
model2=GaussianNB()
model3=SVC()
model4=DecisionTreeClassifier(criterion='entropy')
model5=RandomForestClassifier(n_estimators=100)

modellist=[model1,model2,model3,model4,model5]
```

```
from sklearn.metrics import confusion_matrix,accuracy_score,classification_report

for i in modellist:
    i.fit(x_train,y_train)
    y_pred=i.predict(x_test)
    print('the classification details of model',i,'is below')
    print('the confusion matrix of ',i,'is')
    print( confusion_matrix(y_test,y_pred))
    print('accuracy score of',i,'is')
    print(accuracy_score(y_test,y_pred))
    print('the classification report of',i,'is')
    print(classification_report(y_test,y_pred))
```

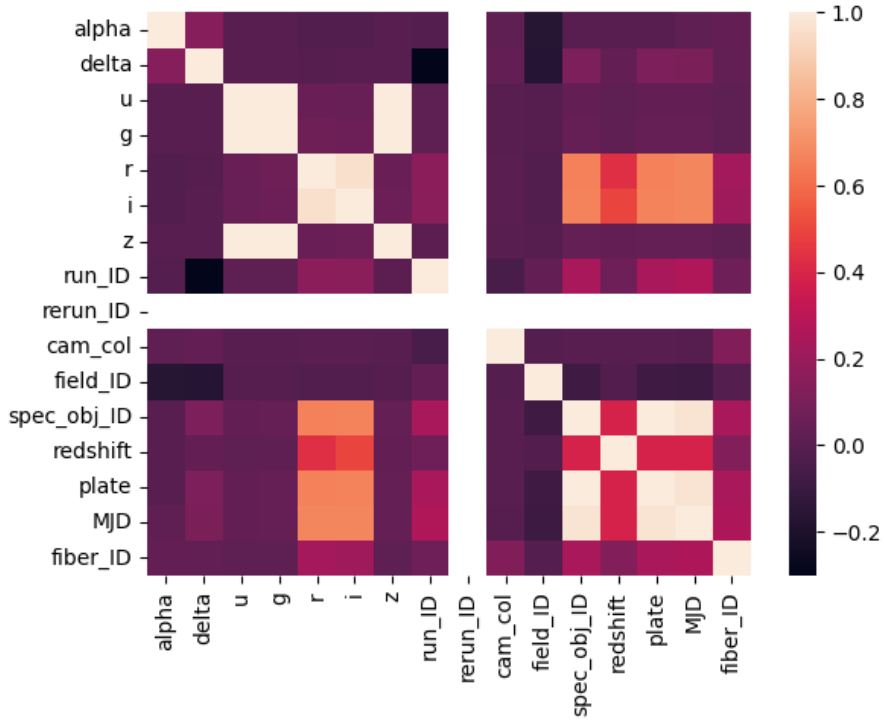
```
    accuracy                           0.84     30000
   macro avg       0.83      0.87      0.83     30000
weighted avg       0.88      0.84      0.85     30000

the classification details of model SVC() is below
the confusion matrix of  SVC() is
[[13857   614  3374]
 [  578  5093    29]
 [    3     0  6452]]
accuracy score of SVC() is
0.8467333333333333
the classification report of SVC() is
              precision    recall  f1-score   support

      GALAXY       0.96      0.78      0.86     17845
         QSO       0.89      0.89      0.89      5700
        STAR       0.65      1.00      0.79      6455

    accuracy                           0.85     30000
   macro avg       0.84      0.89      0.85     30000
weighted avg       0.88      0.85      0.85     30000

the classification details of model DecisionTreeClassifier(criterion='entropy') is below
the confusion matrix of  DecisionTreeClassifier(criterion='entropy') is
[[15513  2175   157]
 [ 1219  4480     1]
 [   21     1  6433]]
accuracy score of DecisionTreeClassifier(criterion='entropy') is
0.8808666666666667
the classification report of DecisionTreeClassifier(criterion='entropy') is
              precision    recall  f1-score   support

      GALAXY       0.93      0.87      0.90     17845
         QSO       0.67      0.79      0.73      5700
        STAR       0.98      1.00      0.99      6455

    accuracy                           0.88     30000
   macro avg       0.86      0.88      0.87     30000
weighted avg       0.89      0.88      0.88     30000

the classification details of model RandomForestClassifier() is below
the confusion matrix of  RandomForestClassifier() is
[[17549   139   157]
 [  953  4746     1]
 [    0     0  6455]]
accuracy score of RandomForestClassifier() is
0.9583333333333334
the classification report of RandomForestClassifier() is
              precision    recall  f1-score   support

      GALAXY       0.95      0.98      0.97     17845
         QSO       0.97      0.83      0.90      5700
        STAR       0.98      1.00      0.99      6455

    accuracy                           0.96     30000
   macro avg       0.97      0.94      0.95     30000
weighted avg       0.96      0.96      0.96     30000
```

```
df.corr()
```

	alpha	delta	u	g	r	i	z	run_ID	rerun_ID	cam_col
alpha	1.000000	0.138691	-0.001532	-0.002423	-0.022083	-0.023580	-0.002918	-0.013737	NaN	0.019582
delta	0.138691	1.000000	0.002074	0.003523	-0.006835	-0.004480	0.003630	-0.301238	NaN	0.032565
u	-0.001532	0.002074	1.000000	0.999311	0.054149	0.045730	0.998093	0.015309	NaN	0.003548
g	-0.002423	0.003523	0.999311	1.000000	0.062387	0.056271	0.999161	0.015710	NaN	0.003508
r	-0.022083	-0.006835	0.054149	0.062387	1.000000	0.962868	0.053677	0.153889	NaN	0.008480
i	-0.023580	-0.004480	0.045730	0.056271	0.962868	1.000000	0.055994	0.147668	NaN	0.007615
z	-0.002918	0.003630	0.998093	0.999161	0.053677	0.055994	1.000000	0.013811	NaN	0.003365
run_ID	-0.013737	-0.301238	0.015309	0.015710	0.153889	0.147668	0.013811	1.000000	NaN	-0.047098
rerun_ID	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
cam_col	0.019582	0.032565	0.003548	0.003508	0.008480	0.007615	0.003365	-0.047098	NaN	1.000000
field_ID	-0.165577	-0.173416	-0.008374	-0.008852	-0.026423	-0.026679	-0.008903	0.031498	NaN	-0.015684
spec_obj_ID	-0.002553	0.112329	0.029997	0.039443	0.655245	0.661641	0.037813	0.239460	NaN	-0.001946
redshift	0.001667	0.031638	0.014309	0.022954	0.433241	0.492383	0.030380	0.065400	NaN	0.000097
plate	-0.002554	0.112329	0.029997	0.039443	0.655243	0.661640	0.037813	0.239459	NaN	-0.001949
MJD	0.019943	0.107333	0.031997	0.040274	0.671180	0.672523	0.037469	0.262687	NaN	-0.006745
fiber_ID	0.030464	0.028250	0.016305	0.017470	0.223106	0.214787	0.014668	0.067165	NaN	0.121597

```
sns.heatmap(df.corr())
```

```
<Axes: >
```




```
ig, axs = plt.subplots(2, 2, figsize=(12, 12))

sns.boxplot(x=df["class"], y=df["alpha"], ax=axs[0, 0])
sns.boxplot(x=df["class"], y=df["delta"], ax=axs[0, 1])
sns.boxplot(x=df["class"], y=df["plate"], ax=axs[1, 0])
sns.boxplot(x=df["class"], y=df["run_ID"], ax=axs[1, 1])
```

```
<Axes: xlabel='class', ylabel='run_ID'>
```