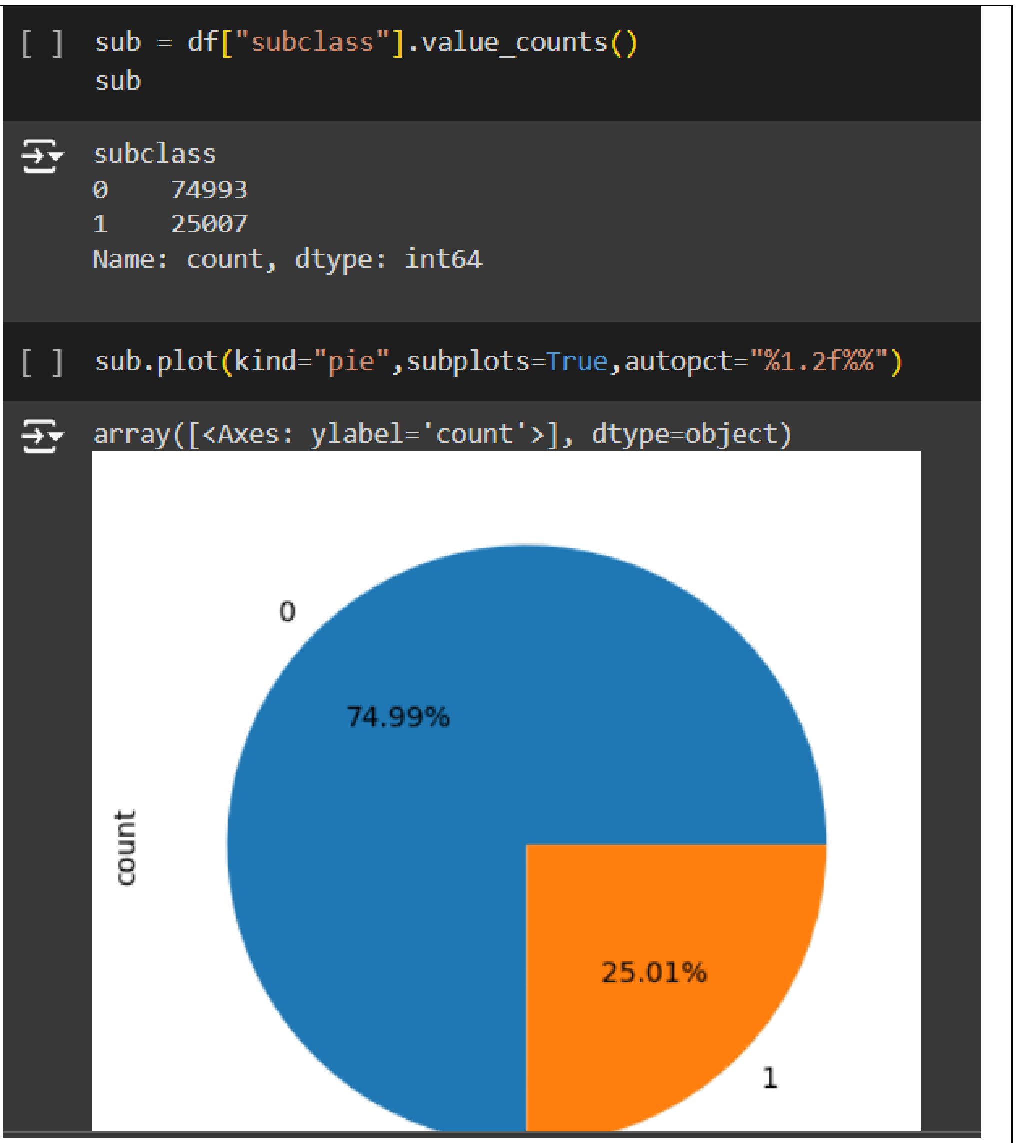
Data Collection and Preprocessing Phase

Date	15 July 2024
Team ID	740051
Project Title	**SDSS galaxy classification using Machine Learning**
Maximum Marks	6 Marks

Data Exploration and Preprocessing Template

Exploration and Preprocessing Template for SDSS galaxy classification for Machine Learning: Load data, handle missing values, explore basic statistics, visualize distributions, encode categorical variables, normalize/scale features, identify outliers, and prepare for modeling

Section	**Description**
Data Overview	Summary of the dataset, including number of rows and columns, data types of each column, and brief descriptions of each column.
Univariate Analysis	Distribution analysis of individual variables using histograms, bar charts, and descriptive statistics (mean, median, mode, standard deviation). #**Univariate Analysis**

<table>
<tr>
<td></td>
<td>


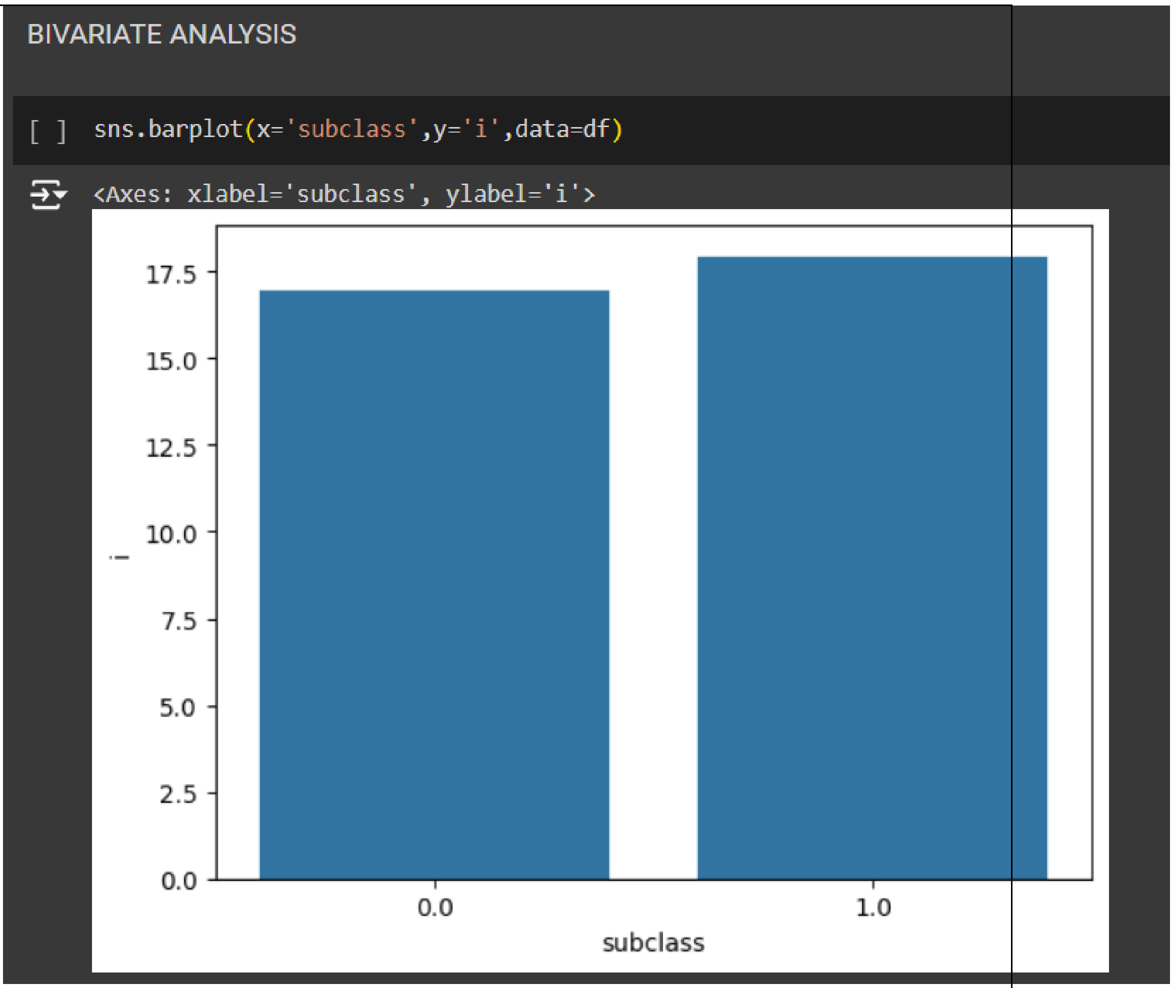
```
[ ]  sub = df["subclass"].value_counts()
     sub
```

```
subclass
0    74993
1    25007
Name: count, dtype: int64
```

```
[ ]  sub.plot(kind="pie",subplots=True,autopct="%1.2f%%")
```

```
array([<Axes: ylabel='count'>], dtype=object)
```

</td>
</tr>
<tr>
<td>Bivariate Analysis</td>
<td>Examination of relationships between pairs of variables using scatter plots, correlation matrices, and pairwise plots to identify patterns and trends.
#<b>Bivariate Analysis</b></td>
</tr>
</table>



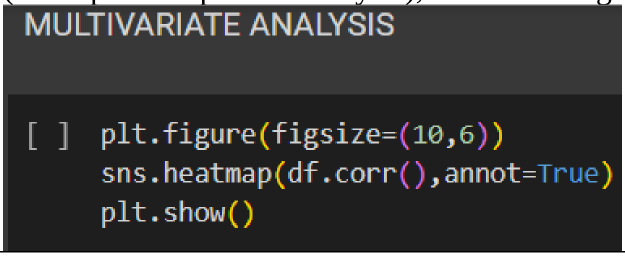
BIVARIATE ANALYSIS

```
[ ]  sns.barplot(x='subclass',y='i',data=df)
```

```
<Axes: xlabel='subclass', ylabel='i'>
```

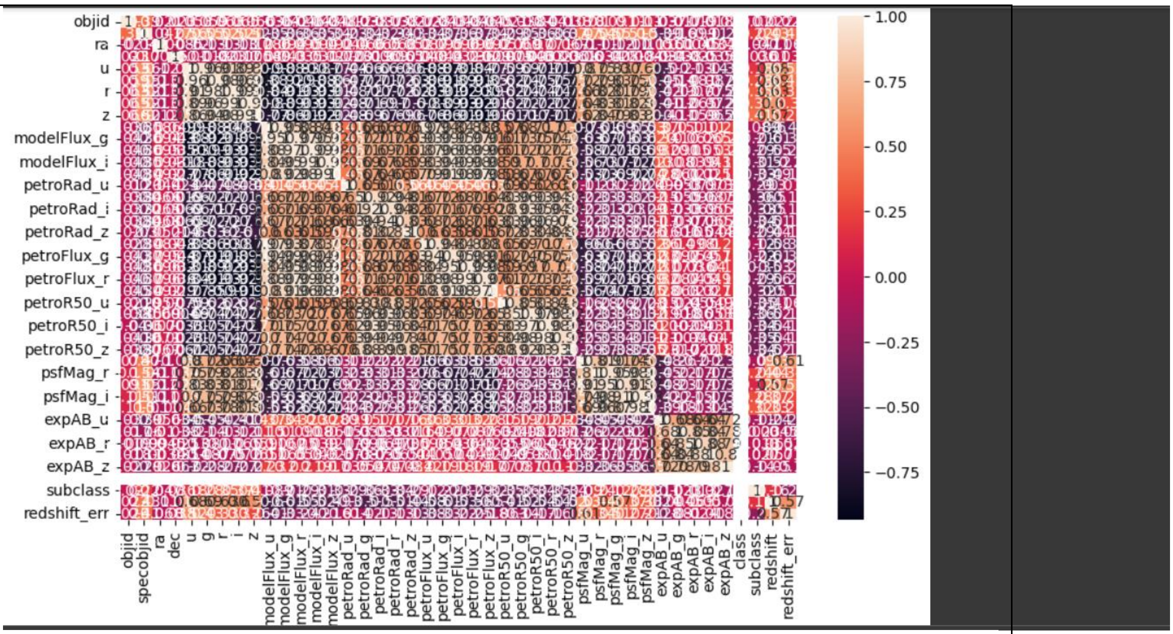
Multivariate Analysis

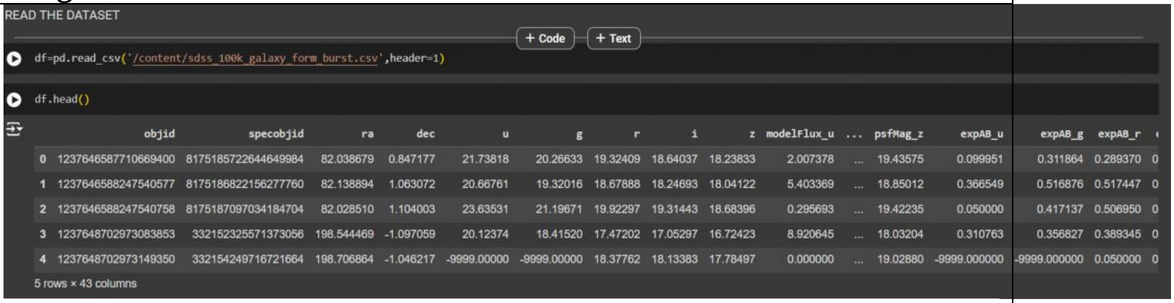
Investigation of interactions between multiple variables using heatmaps, PCA (Principal Component Analysis), and clustering to understand data structure.

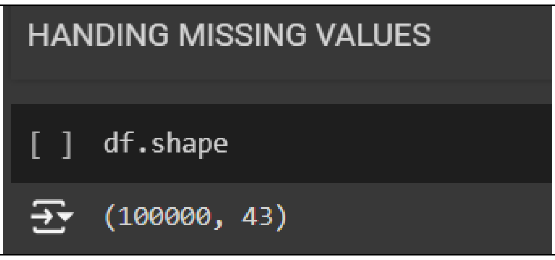


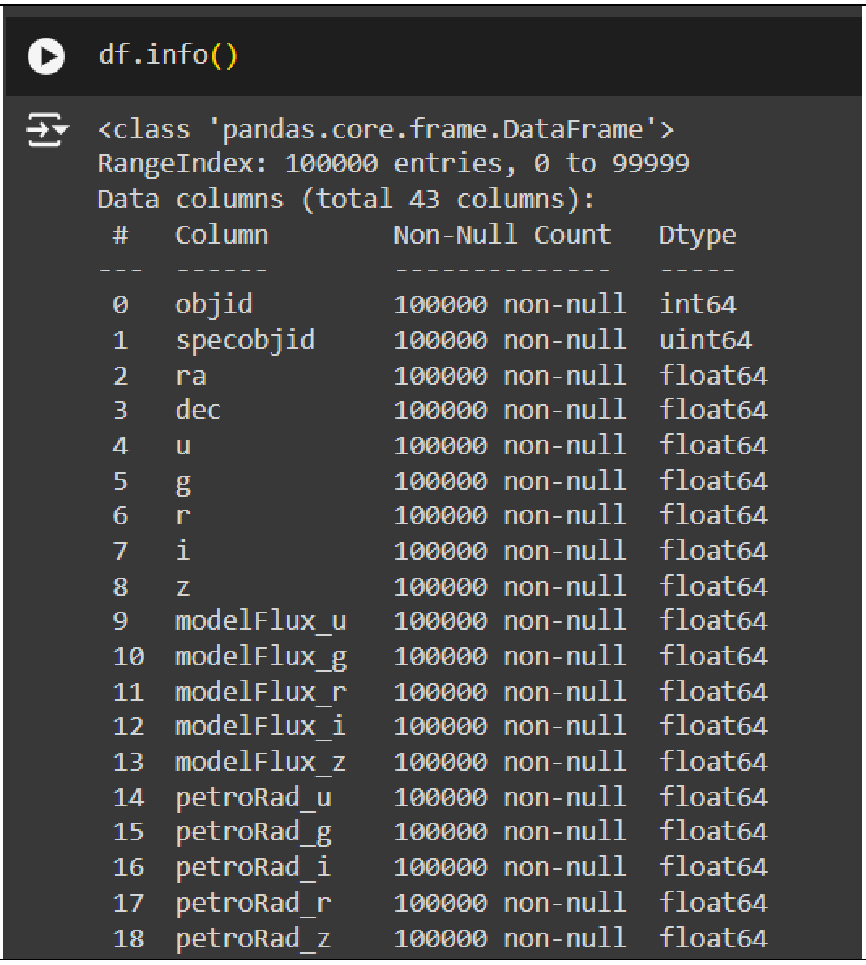
MULTIVARIATE ANALYSIS

```
[ ]  plt.figure(figsize=(10,6))
     sns.heatmap(df.corr(),annot=True)
     plt.show()
```

	
Outliers and Anomalies	Identification and description of outliers and anomalies, summarized in a table with details on detection method, number of outliers, description, and potential impact.
Data Preprocessing Code Screenshots	

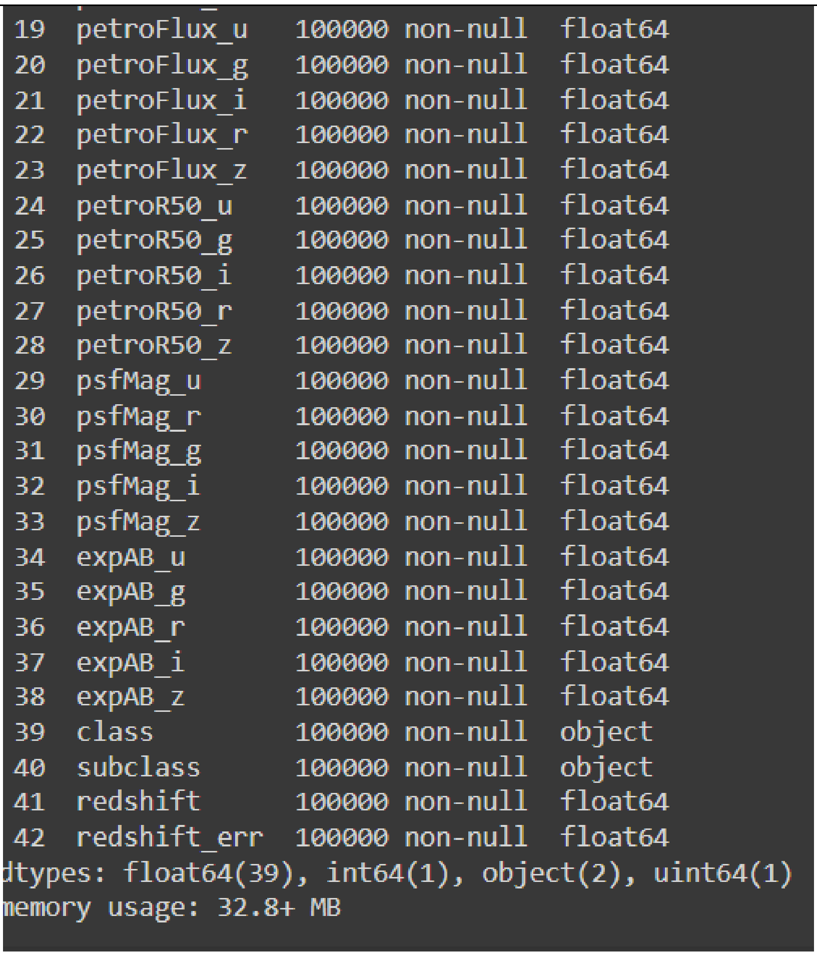
Loading Data	READ THE DATASET
	`df=pd.read_csv('/content/sdss_100k_galaxy_form_burst.csv',header=1)`
	`df.head()`
	`objid specobjid ra dec u g r i z modelFlux_u ... psfMag_z expAB_u expAB_g expAB_r`
	`0 1237646587710669400 8175185722644649984 82.038679 0.847177 21.73818 20.26633 19.32409 18.64037 18.23833 2.007378 ... 19.43575 0.099951 0.311864 0.289370 0`
	`1 1237646588247540577 8175186822156277760 82.138894 1.063072 20.66761 19.32016 18.67888 18.24693 18.04122 5.403369 ... 18.85012 0.366549 0.516876 0.517447 0`
	`2 1237646588247540758 8175187097034184704 82.028510 1.104003 23.63531 21.19671 19.92297 19.31443 18.68396 0.295693 ... 19.42235 0.050000 0.417137 0.506950 0`
	`3 1237648702973083853 332152325571373056 198.544469 -1.097059 20.12374 18.41520 17.47202 17.05297 16.72423 8.920645 ... 18.03204 0.310763 0.356827 0.389345 0`
	`4 1237648702973149350 332154249716721664 198.706864 -1.046217 -9999.00000 -9999.00000 18.37762 18.13383 17.78497 0.000000 ... 19.02880 -9999.000000 -9999.000000 0.050000 0`
	5 rows × 43 columns

Handling Missing Data	HANDING MISSING VALUES
	`[ ] df.shape`
	`(100000, 43)`



```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 43 columns):
 #   Column        Non-Null Count   Dtype
---  ------        --------------   -----
 0   objid         100000 non-null  int64
 1   specobjid     100000 non-null  uint64
 2   ra            100000 non-null  float64
 3   dec           100000 non-null  float64
 4   u             100000 non-null  float64
 5   g             100000 non-null  float64
 6   r             100000 non-null  float64
 7   i             100000 non-null  float64
 8   z             100000 non-null  float64
 9   modelFlux_u   100000 non-null  float64
 10  modelFlux_g   100000 non-null  float64
 11  modelFlux_r   100000 non-null  float64
 12  modelFlux_i   100000 non-null  float64
 13  modelFlux_z   100000 non-null  float64
 14  petroRad_u    100000 non-null  float64
 15  petroRad_g    100000 non-null  float64
 16  petroRad_i    100000 non-null  float64
 17  petroRad_r    100000 non-null  float64
 18  petroRad_z    100000 non-null  float64
```

	 19 petroFlux_u 100000 non-null float64 20 petroFlux_g 100000 non-null float64 21 petroFlux_i 100000 non-null float64 22 petroFlux_r 100000 non-null float64 23 petroFlux_z 100000 non-null float64 24 petroR50_u 100000 non-null float64 25 petroR50_g 100000 non-null float64 26 petroR50_i 100000 non-null float64 27 petroR50_r 100000 non-null float64 28 petroR50_z 100000 non-null float64 29 psfMag_u 100000 non-null float64 30 psfMag_r 100000 non-null float64 31 psfMag_g 100000 non-null float64 32 psfMag_i 100000 non-null float64 33 psfMag_z 100000 non-null float64 34 expAB_u 100000 non-null float64 35 expAB_g 100000 non-null float64 36 expAB_r 100000 non-null float64 37 expAB_i 100000 non-null float64 38 expAB_z 100000 non-null float64 39 class 100000 non-null object 40 subclass 100000 non-null object 41 redshift 100000 non-null float64 42 redshift_err 100000 non-null float64 dtypes: float64(39), int64(1), object(2), uint64(1) memory usage: 32.8+ MB For checking the null values, . isnull() function is used. To sum those null values we use . sum() function. From the above image we found that there are no null values present in our dataset. So we can skip handling the missing values step.
Data Transformation	-
Feature Engineering	-
Save Processed Data	-