Problem Statement

In the galaxy's far reaches, the United Intergalactic Council (UIC) has undertaken an ambitious project to catalog all the stars in the known universe. As part of this project, a dataset has been collected containing various features of stars, such as absolute temperature, relative luminosity, relative radius, absolute magnitude, star color, spectral class, and star type.

The UIC has categorized the stars into six unique and fascinating types (fictional names):

Crimson Dwarfs Umber Dwarfs Pearl Dwarfs Aurelian Mainstays Celestial Sovereigns Cosmic Behemoths
The UIC calls upon the brightest minds to develop a machine-learning model to predict these extraordinary star types based on the given features. As the star catalog grows, the UIC wants to ensure its classification system remains accurate and efficient.

In [1]:
```
# Importing Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
# Loading Dataset
df = pd.read_csv("train.csv")
```

In [3]:
```
df.head()
```

Out[3]:

	Temperature (K)	Luminosity(L/Lo)	Radius(R/Ro)	Absolute magnitude(Mv)	Star type	Star color	Spectral Class
0	3068	0.002400	0.1700	16.12	Umber Dwarfs	Red	M
1	3042	0.000500	0.1542	16.60	Umber Dwarfs	Red	M
2	2600	0.000300	0.1020	18.70	Umber Dwarfs	Red	M
3	2800	0.000200	0.1600	16.65	Umber Dwarfs	Red	M
4	1939	0.000138	0.1030	20.06	Umber Dwarfs	Red	M

In [4]:
```
df.shape
```

Out[4]:
```
(216, 7)
```

In [5]:
```
df.isnull().sum()
```

Out[5]:
```
Temperature (K)           0
Luminosity(L/Lo)          0
Radius(R/Ro)              0
Absolute magnitude(Mv)    0
Star type                 0
Star color                0
Spectral Class            0
dtype: int64
```

In [6]:
```
df.duplicated().sum()
```

Out[6]:
```
0
```

In [7]:
```
df['Star type'].value_counts()/216 *100
```

Out[7]:
```
Crimson Dwarfs          17.592593
Aurelian Mainstays      17.129630
Pearl Dwarfs            16.666667
Celestial Sovereigns    16.666667
Umber Dwarfs            16.203704
Cosmic Behemoths        15.740741
Name: Star type, dtype: float64
```

In [8]:
```
df['Star color'].value_counts()/216 *100
```

Out[8]:
```
Red                   47.222222
Blue                  22.685185
Blue-white            10.648148
yellow-white           3.703704
White                  3.240741
Blue White             3.240741
Yellowish White        1.388889
Blue white             1.388889
white                  1.388889
Whitish                0.925926
yellowish              0.925926
Pale yellow orange     0.462963
Orange                 0.462963
White-Yellow           0.462963
Blue                   0.462963
Orange-Red             0.462963
Blue white             0.462963
Blue-White             0.462963
Name: Star color, dtype: float64
```

In [9]:
```
df['Star color'].value_counts().index
```

Out[9]:
```
Index(['Red', 'Blue', 'Blue-white', 'yellow-white', 'White', 'Blue White',
       'Yellowish White', 'Blue white', 'white', 'Whitish', 'yellowish',
       'Pale yellow orange', 'Orange', 'White-Yellow', 'Blue ', 'Orange-Red',
       'Blue white ', 'Blue-White'],
      dtype='object')
```

In [10]:
```
df['Spectral Class'].value_counts()/216 *100
```

Out[10]:
```
M    46.759259
B    18.518519
O    16.203704
A     8.333333
F     7.870370
K     1.851852
G     0.462963
Name: Spectral Class, dtype: float64
```

In [11]:
```
df.shape
```

Out[11]:
```
(216, 7)
```

In [12]:
```
df.loc[df['Spectral Class'].isin(['K','G']), 'Spectral Class'] = 'others'
```

In [13]:
```
df['Spectral Class'].value_counts()
```

Out[13]:
```
M         101
B          40
O          35
A          18
F          17
others      5
Name: Spectral Class, dtype: int64
```

In [14]:
```
star_color = ['yellow-white', 'White', 'Blue White','Yellowish White', 'Blue white', 'whi
              'Pale yellow orange', 'Orange', 'White-Yellow', 'Blue ', 'Orange-Red',
              'Blue white ', 'Blue-White']

df.loc[df['Star color'].isin(star_color), 'Star color'] = 'others'
```

In [15]:
```
df['Star color'].value_counts()
```

Out[15]:
```
Red           102
Blue           49
others         42
Blue-white     23
Name: Star color, dtype: int64
```

In [16]:
```
df.head()
```

Out[16]:

	Temperature (K)	Luminosity(L/Lo)	Radius(R/Ro)	Absolute magnitude(Mv)	Star type	Star color	Spectral Class
0	3068	0.002400	0.1700	16.12	Umber Dwarfs	Red	M
1	3042	0.000500	0.1542	16.60	Umber Dwarfs	Red	M
2	2600	0.000300	0.1020	18.70	Umber Dwarfs	Red	M
3	2800	0.000200	0.1600	16.65	Umber Dwarfs	Red	M
4	1939	0.000138	0.1030	20.06	Umber Dwarfs	Red	M

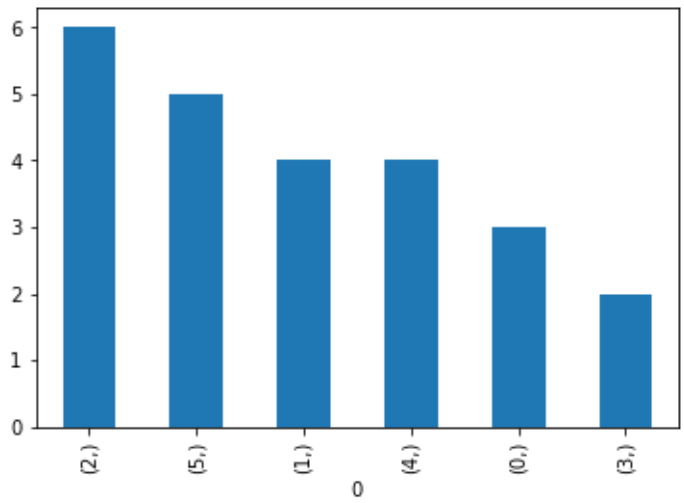
XGBoost

In [17]:
```
import xgboost as xgb
from sklearn import preprocessing
X = df.drop(columns='Star type')
y = df['Star type']

print(pd.DataFrame(y).value_counts())

lbl = preprocessing.LabelEncoder()
X['Star color'] = lbl.fit_transform(X['Star color'].astype(str))
X['Spectral Class'] = lbl.fit_transform(X['Spectral Class'].astype(str))
y = lbl.fit_transform(y.astype(str))

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25,
                                                    random_state=1)
xgbr = xgb.XGBClassifier(n_estimators=1000,max_depth=10)

print(pd.DataFrame(y).value_counts())

xgbr.fit(X_train, y_train)
y_pred = xgbr.predict(X_test)

from sklearn.metrics import classification_report,accuracy_score
print("Accuracy Score : ",accuracy_score(y_test,y_pred))
print("***** classification_report *****")
print(classification_report(y_test,y_pred))
```

```
Star type
Crimson Dwarfs          38
Aurelian Mainstays      37
Celestial Sovereigns    36
Pearl Dwarfs            36
Umber Dwarfs            35
Cosmic Behemoths        34
dtype: int64
3    38
0    37
1    36
4    36
5    35
2    34
dtype: int64
Accuracy Score :  0.9814814814814815
***** classification_report *****
              precision    recall  f1-score   support

           0       1.00      0.88      0.93         8
           1       1.00      1.00      1.00        12
           2       1.00      1.00      1.00        10
           3       0.90      1.00      0.95         9
           4       1.00      1.00      1.00         7
           5       1.00      1.00      1.00         8

    accuracy                           0.98        54
   macro avg       0.98      0.98      0.98        54
weighted avg       0.98      0.98      0.98        54
```

In [18]:
```
test = pd.read_csv('test.csv')

lbl = preprocessing.LabelEncoder()
test['Star color'] = lbl.fit_transform(test['Star color'].astype(str))
test['Spectral Class'] = lbl.fit_transform(test['Spectral Class'].astype(str))

y_pred1 = xgbr.predict(test)
pd.DataFrame(y_pred1).value_counts().plot(kind='bar')
```

Out[18]:
```
<AxesSubplot:xlabel='0'>
```



In [19]:
```
pd.DataFrame(y_pred1).value_counts()
```

Out[19]:
```
2    6
5    5
1    4
4    4
0    3
3    2
dtype: int64
```

In [20]:
```
# Crimson Dwarfs          3
# Aurelian Mainstays      0
# Celestial Sovereigns    1
# Pearl Dwarfs            4
# Umber Dwarfs            5
# Cosmic Behemoths        2
```

In [21]:
```
df.loc[df['Spectral Class'].isin(['K','G']), 'Spectral Class'] = 'others'
```

In [22]:
```
sol = pd.DataFrame(y_pred1)
sol.loc[sol[0].isin([3]), 0] = 'Crimson Dwarfs'
sol.loc[sol[0].isin([0]), 0] = 'Aurelian Mainstays'
sol.loc[sol[0].isin([1]), 0] = 'Celestial Sovereigns'
sol.loc[sol[0].isin([4]), 0] = 'Pearl Dwarfs'
sol.loc[sol[0].isin([5]), 0] = 'Umber Dwarfs'
sol.loc[sol[0].isin([2]), 0] = 'Cosmic Behemoths'
```

In [23]:
```
sol.value_counts()
```

Out[23]:
```
Cosmic Behemoths        6
Umber Dwarfs            5
Celestial Sovereigns    4
Pearl Dwarfs            4
Aurelian Mainstays      3
Crimson Dwarfs          2
dtype: int64
```

In [24]:
```
sol
```

Out[24]:

	0
0	Pearl Dwarfs
1	Umber Dwarfs
2	Aurelian Mainstays
3	Celestial Sovereigns
4	Celestial Sovereigns
5	Cosmic Behemoths
6	Cosmic Behemoths
7	Pearl Dwarfs
8	Umber Dwarfs
9	Pearl Dwarfs
10	Celestial Sovereigns
11	Pearl Dwarfs
12	Cosmic Behemoths
13	Celestial Sovereigns
14	Cosmic Behemoths
15	Cosmic Behemoths
16	Umber Dwarfs
17	Crimson Dwarfs
18	Aurelian Mainstays
19	Umber Dwarfs
20	Umber Dwarfs
21	Crimson Dwarfs
22	Cosmic Behemoths
23	Aurelian Mainstays