In [1]:

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

In [2]:

```
mydata=pd.read_csv("C:Downloads/xAPI-Edu-Data.csv")
```

In [3]:

```
mydata
```

Out[3]:

| | gender | NationalITy | PlaceofBirth | StageID | GradeID | SectionID | Topic | Semester |
|---|---|---|---|---|---|---|---|---|
| **0** | M | KW | KuwaIT | lowerlevel | G-04 | A | IT | F |
| **1** | M | KW | KuwaIT | lowerlevel | G-04 | A | IT | F |
| **2** | M | KW | KuwaIT | lowerlevel | G-04 | A | IT | F |
| **3** | M | KW | KuwaIT | lowerlevel | G-04 | A | IT | F |
| **4** | M | KW | KuwaIT | lowerlevel | G-04 | A | IT | F |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... |
| **475** | F | Jordan | Jordan | MiddleSchool | G-08 | A | Chemistry | S |
| **476** | F | Jordan | Jordan | MiddleSchool | G-08 | A | Geology | F |
| **477** | F | Jordan | Jordan | MiddleSchool | G-08 | A | Geology | S |
| **478** | F | Jordan | Jordan | MiddleSchool | G-08 | A | History | F |
| **479** | F | Jordan | Jordan | MiddleSchool | G-08 | A | History | S |

480 rows × 17 columns

In [4]:

```
mydata.isnull().sum()
```

Out[4]:

```
gender                      0
NationalITy                 0
PlaceofBirth                0
StageID                     0
GradeID                     0
SectionID                   0
Topic                       0
Semester                    0
Relation                    0
raisedhands                 0
VisITedResources            0
AnnouncementsView           0
Discussion                  0
ParentAnsweringSurvey       0
ParentschoolSatisfaction    0
StudentAbsenceDays          0
Class                       0
dtype: int64
```

In [5]:

```
mydata.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 480 entries, 0 to 479
Data columns (total 17 columns):
 #   Column                    Non-Null Count  Dtype 
---  ------                    --------------  ----- 
 0   gender                    480 non-null    object
 1   NationalITy               480 non-null    object
 2   PlaceofBirth              480 non-null    object
 3   StageID                   480 non-null    object
 4   GradeID                   480 non-null    object
 5   SectionID                 480 non-null    object
 6   Topic                     480 non-null    object
 7   Semester                  480 non-null    object
 8   Relation                  480 non-null    object
 9   raisedhands               480 non-null    int64 
 10  VisITedResources          480 non-null    int64 
 11  AnnouncementsView         480 non-null    int64 
 12  Discussion                480 non-null    int64 
 13  ParentAnsweringSurvey     480 non-null    object
 14  ParentschoolSatisfaction  480 non-null    object
 15  StudentAbsenceDays        480 non-null    object
 16  Class                     480 non-null    object
dtypes: int64(4), object(13)
memory usage: 63.9+ KB
```

In [6]:

```
col_to_use=['gender', 'StageID', 'GradeID', 'SectionID', 'Topic',
            'Semester', 'Relation', 'raisedhands', 'VisITedResources',
            'AnnouncementsView', 'Discussion', 'ParentAnsweringSurvey',
            'StudentAbsenceDays', 'ParentschoolSatisfaction', 'Class']
```

In [7]:

```
mydata=mydata[col_to_use]
```

In [8]:

```
mydata.describe()
```

Out[8]:

| | raisedhands | VisITedResources | AnnouncementsView | Discussion |
|---|---|---|---|---|
| **count** | 480.000000 | 480.000000 | 480.000000 | 480.000000 |
| **mean** | 46.775000 | 54.797917 | 37.918750 | 43.283333 |
| **std** | 30.779223 | 33.080007 | 26.611244 | 27.637735 |
| **min** | 0.000000 | 0.000000 | 0.000000 | 1.000000 |
| **25%** | 15.750000 | 20.000000 | 14.000000 | 20.000000 |
| **50%** | 50.000000 | 65.000000 | 33.000000 | 39.000000 |
| **75%** | 75.000000 | 84.000000 | 58.000000 | 70.000000 |
| **max** | 100.000000 | 99.000000 | 98.000000 | 99.000000 |

# Data Visualization

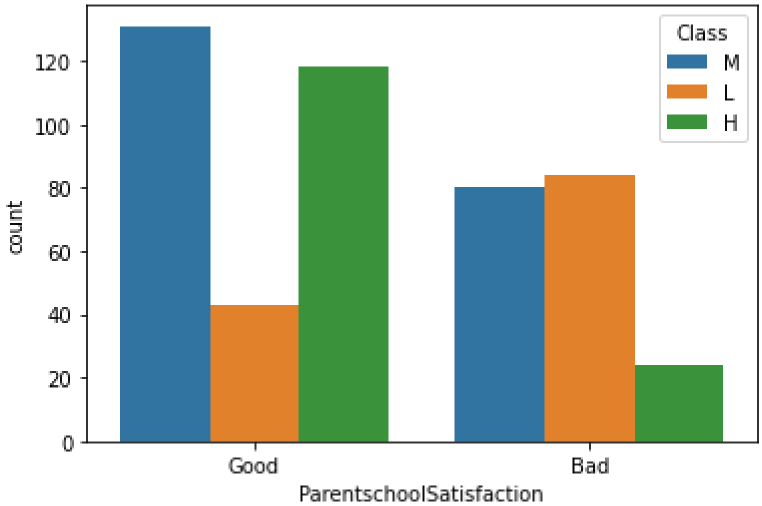
In [9]:

```
sns.countplot(x="gender",data=mydata,hue="Class");
plt.title("Class based on Gender");
```


From the above graph, we can relate the number of male and female present in the class.

In [10]:

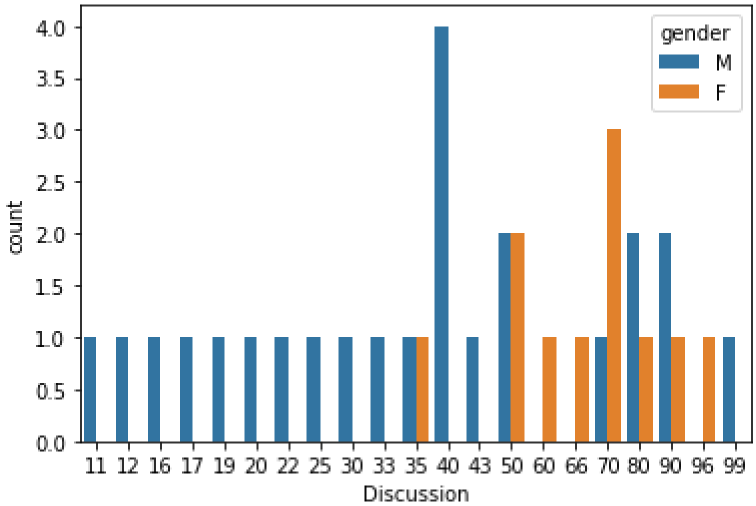
```
sns.countplot(x="ParentschoolSatisfaction",data=mydata,hue="Class");
```




From the above graph, Parent school satisfaction is good for M and H. But it is bad for M and L.

In [11]:

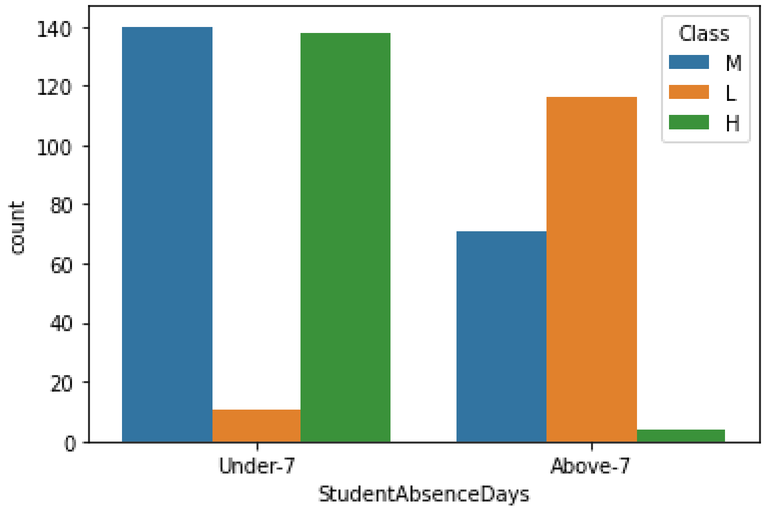
```
sns.countplot(x="Discussion", hue="gender", data=mydata.head(35));
```




From the above graph, Male discussion is higher than female discussion by 4.0

In [12]:

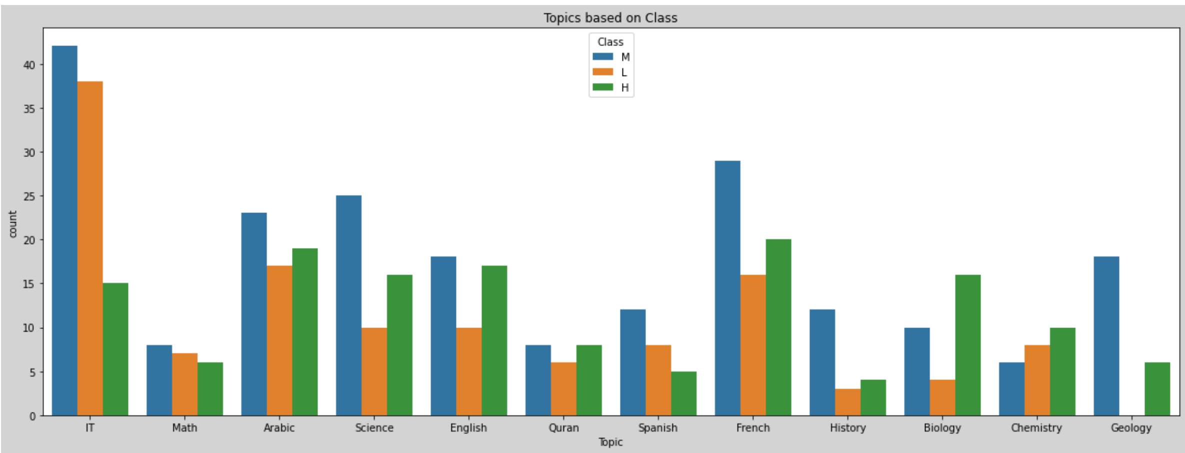
```
sns.countplot(x="StudentAbsenceDays", data=mydata, hue="Class");
```




From the above graph, We can see that the student absent days under-7 for M and H class is more. Similarly the student absent days above-7 for M and L are more.

In [13]:

```
plt.figure(figsize=(20,7),facecolor="lightgrey",frameon=True,edgecolor='blue');
sns.countplot(x="Topic", data=mydata, hue="Class");
plt.title("Topics based on Class");
```



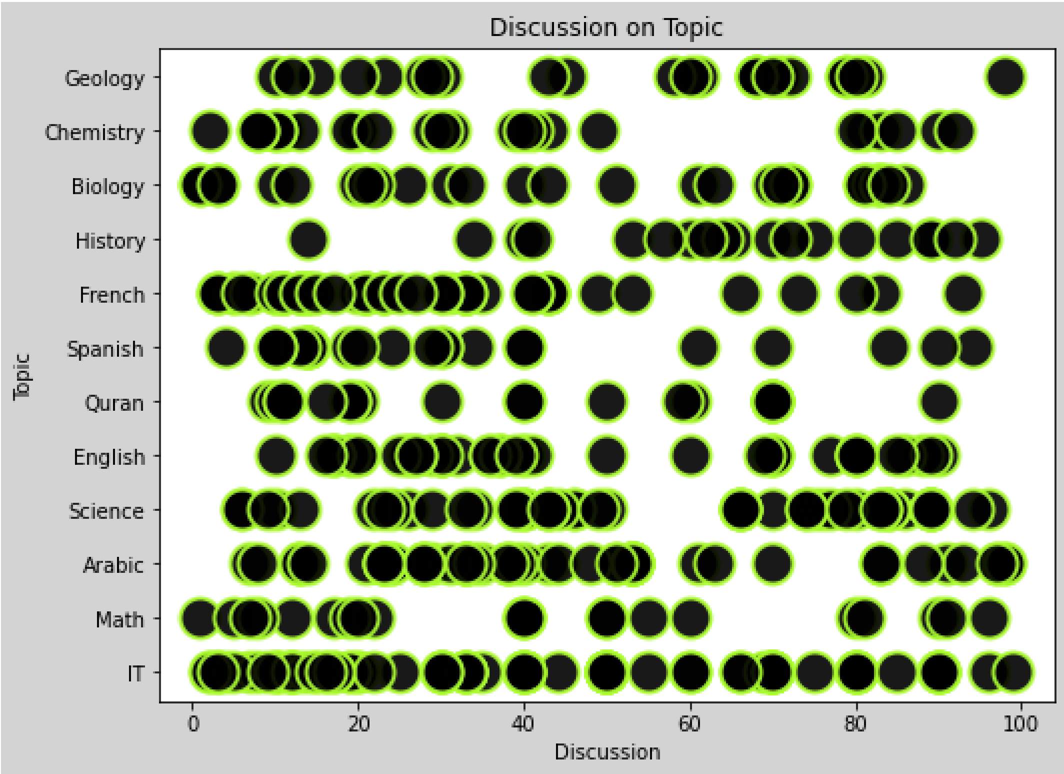

From the above graph we can see that IT is the most dominating topic based on class.

In [14]:

```
plt.figure(figsize=(8,6),facecolor="lightgrey")
plt.scatter(mydata.Discussion,mydata.Topic,
            color="black",alpha=0.9,edgecolors="greenyellow",linewidths=2,s=400)
plt.xlabel("Discussion")
plt.ylabel("Topic")
plt.title("Discussion on Topic")
```

Out[14]:

```
Text(0.5, 1.0, 'Discussion on Topic')
```




In [15]:

```
mydata_corr=mydata.corr()
mydata_corr
```

Out[15]:

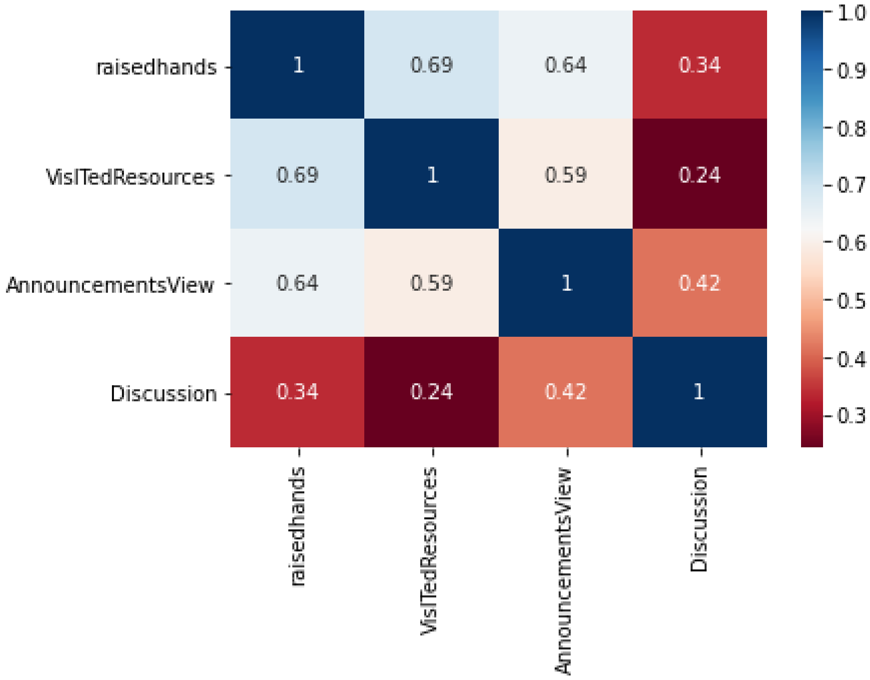
| | raisedhands | VisITedResources | AnnouncementsView | Discussion |
|---|---|---|---|---|
| **raisedhands** | 1.000000 | 0.691572 | 0.643918 | 0.339386 |
| **VisITedResources** | 0.691572 | 1.000000 | 0.594500 | 0.243292 |
| **AnnouncementsView** | 0.643918 | 0.594500 | 1.000000 | 0.417290 |
| **Discussion** | 0.339386 | 0.243292 | 0.417290 | 1.000000 |

In [16]:

```
sns.heatmap(data=mydata_corr, annot=True, cmap='RdBu')
```

Out[16]:

<AxesSubplot:>




In [17]:

```
#From the above heatmap,
#we can see that the data columns having more than 50% are very well correlated with eachot
```

In [18]:

```
from sklearn.preprocessing import LabelEncoder
```

In [19]:

```
LE=LabelEncoder()
```

In [20]:

```
mydata["gender"]=LE.fit_transform(mydata.gender)
mydata["StageID"]=LE.fit_transform(mydata.StageID)
mydata["GradeID"]=LE.fit_transform(mydata.GradeID)
mydata["SectionID"]=LE.fit_transform(mydata.SectionID)
mydata["Topic"]=LE.fit_transform(mydata.Topic)
mydata["Semester"]=LE.fit_transform(mydata.Semester)
mydata["Relation"]=LE.fit_transform(mydata.Relation)
mydata["ParentAnsweringSurvey"]=LE.fit_transform(mydata.ParentAnsweringSurvey)
mydata["ParentschoolSatisfaction"]=LE.fit_transform(mydata.ParentschoolSatisfaction)
mydata["StudentAbsenceDays"]=LE.fit_transform(mydata.StudentAbsenceDays)
mydata["Class"]=LE.fit_transform(mydata.Class)
```

In [21]:

```
mydata
```

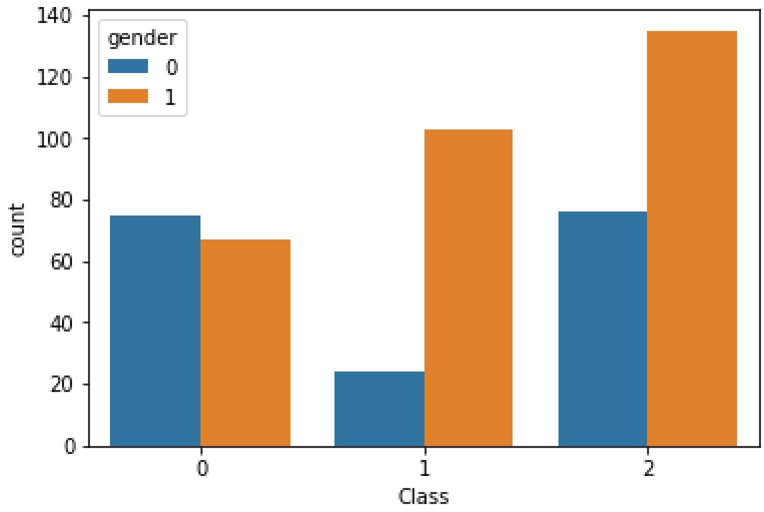
Out[21]:

| | gender | StageID | GradeID | SectionID | Topic | Semester | Relation | raisedhands | VisITedResc |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 2 | 1 | 0 | 7 | 0 | 0 | 15 | |
| **1** | 1 | 2 | 1 | 0 | 7 | 0 | 0 | 20 | |
| **2** | 1 | 2 | 1 | 0 | 7 | 0 | 0 | 10 | |
| **3** | 1 | 2 | 1 | 0 | 7 | 0 | 0 | 30 | |
| **4** | 1 | 2 | 1 | 0 | 7 | 0 | 0 | 40 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | |
| **475** | 0 | 1 | 5 | 0 | 2 | 1 | 0 | 5 | |
| **476** | 0 | 1 | 5 | 0 | 5 | 0 | 0 | 50 | |
| **477** | 0 | 1 | 5 | 0 | 5 | 1 | 0 | 55 | |
| **478** | 0 | 1 | 5 | 0 | 6 | 0 | 0 | 30 | |
| **479** | 0 | 1 | 5 | 0 | 6 | 1 | 0 | 35 | |

480 rows × 15 columns

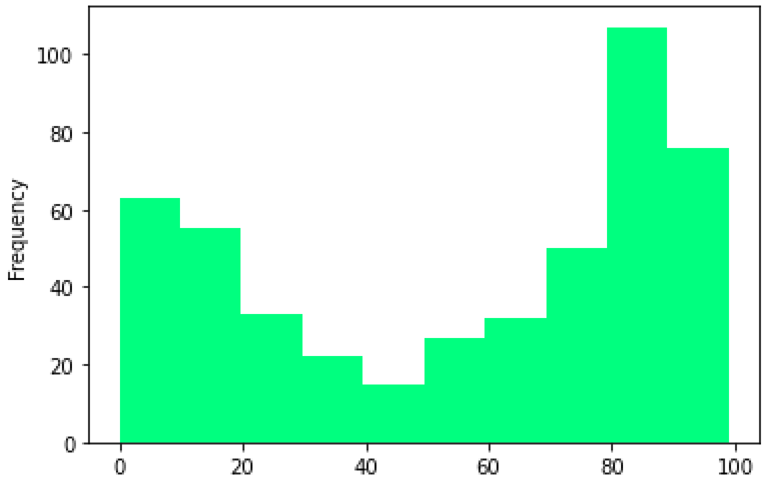
In [22]:

```
sns.countplot(x="Class",data=mydata,hue="gender");
```



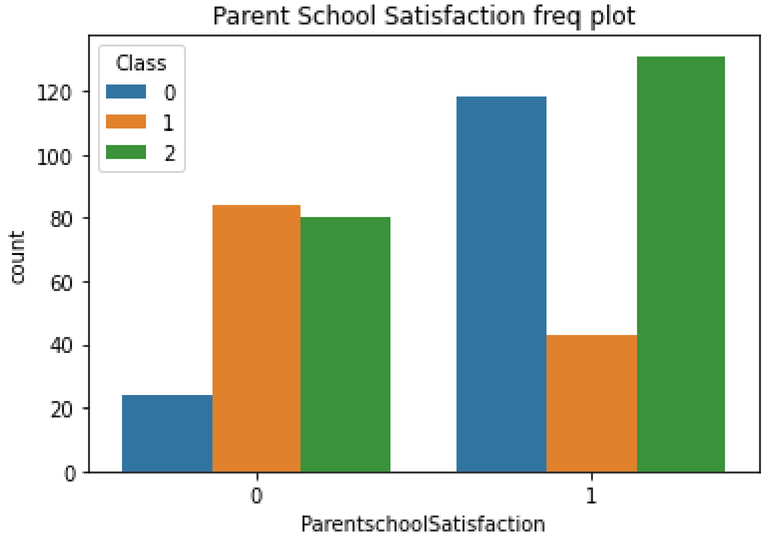

In [23]:

```
mydata.VisITedResources.plot.hist(color="springgreen");
```



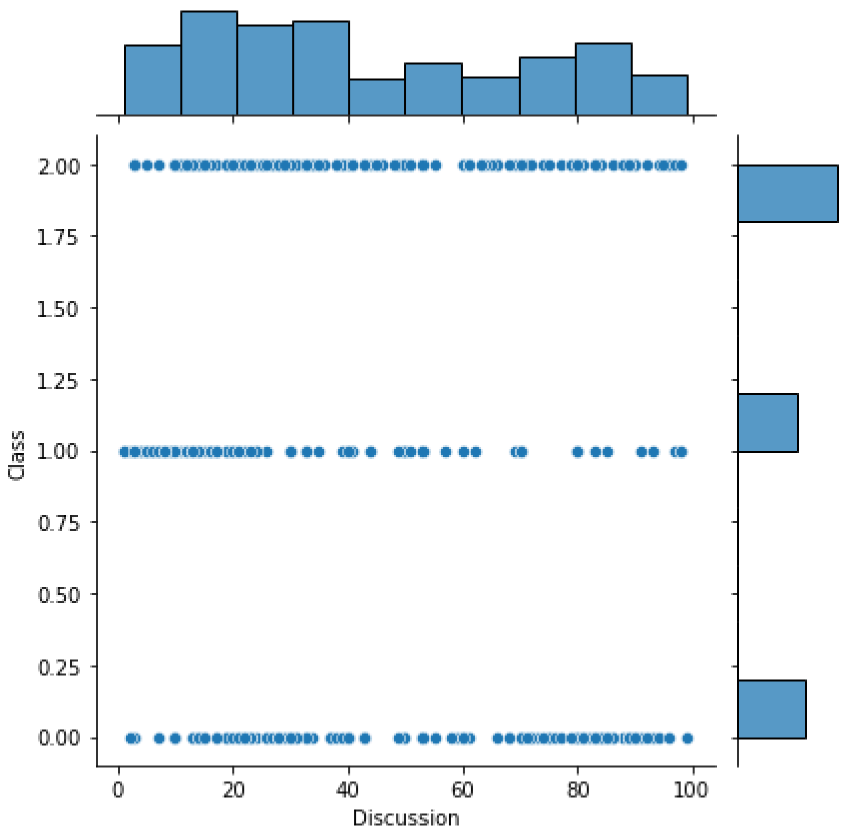
In [24]:

```
sns.countplot(x="ParentschoolSatisfaction",data=mydata,hue="Class");
plt.title("Parent School Satisfaction freq plot");
```




In [38]:

```
sns.jointplot(x="Discussion",y="Class",data=mydata);
```




# Separating dep and indep variables

In [26]:

```
y_dep = mydata.Class
y_dep
```

Out[26]:

```
0      2
1      2
2      1
3      1
4      2
      ..
475    1
476    2
477    2
478    1
479    1
Name: Class, Length: 480, dtype: int32
```

In [27]:

```
x_ind=mydata.drop("Class",axis=1)
x_ind
```

Out[27]:

| | gender | StageID | GradeID | SectionID | Topic | Semester | Relation | raisedhands | VisITedResc |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 2 | 1 | 0 | 7 | 0 | 0 | 15 | |
| **1** | 1 | 2 | 1 | 0 | 7 | 0 | 0 | 20 | |
| **2** | 1 | 2 | 1 | 0 | 7 | 0 | 0 | 10 | |
| **3** | 1 | 2 | 1 | 0 | 7 | 0 | 0 | 30 | |
| **4** | 1 | 2 | 1 | 0 | 7 | 0 | 0 | 40 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | |
| **475** | 0 | 1 | 5 | 0 | 2 | 1 | 0 | 5 | |
| **476** | 0 | 1 | 5 | 0 | 5 | 0 | 0 | 50 | |
| **477** | 0 | 1 | 5 | 0 | 5 | 1 | 0 | 55 | |
| **478** | 0 | 1 | 5 | 0 | 6 | 0 | 0 | 30 | |
| **479** | 0 | 1 | 5 | 0 | 6 | 1 | 0 | 35 | |

480 rows × 14 columns

In [28]:

```
from sklearn.model_selection import train_test_split
```

In [29]:

```
x_train,x_test,y_train,y_test=train_test_split(x_ind,y_dep,test_size=0.2,random_state=2)
```

In [30]:

```
from sklearn.linear_model import LogisticRegression
```

In [31]:

```
modelLR=LogisticRegression()
```

In [32]:

```
modelLR.fit(x_train,y_train)
```

Out[32]:

```
LogisticRegression()
```

In [33]:

```
y_pred=modelLR.predict(x_test)
```

In [34]:

```
y_pred
```

Out[34]:

```
array([0, 0, 0, 2, 0, 0, 1, 1, 2, 2, 1, 1, 0, 2, 1, 2, 0, 0, 2, 0, 0, 0,
       2, 1, 2, 2, 0, 1, 0, 2, 1, 0, 2, 2, 0, 1, 1, 1, 2, 1, 1, 0, 0, 0,
       2, 2, 0, 1, 0, 1, 1, 2, 0, 2, 2, 2, 0, 1, 1, 2, 0, 1, 2, 2, 2, 1,
       1, 2, 2, 0, 0, 1, 2, 2, 2, 0, 2, 1, 2, 2, 2, 2, 1, 1, 1, 2, 1, 0,
       0, 0, 1, 2, 2, 0, 0, 0])
```

# Confusion Matrix and Accuracy Score

In [35]:

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

In [36]:

```
confusion_matrix(y_test,y_pred)
```

Out[36]:

```
array([[20,  0, 14],
       [ 0, 21,  2],
       [12,  7, 20]], dtype=int64)
```

In [37]:

```
accuracy_score(y_test,y_pred)
```

Out[37]:

```
0.6354166666666666
```

In [ ]: