Logistics Regression

In [106…

```
import pandas as pd
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, accuracy_score
```

In [107…

```
data = pd.read_csv('xAPI-Edu-Data.csv')
```

In [108…

```
data.head()
```

Out[108]:

	gender	NationalITy	PlaceofBirth	StageID	GradeID	SectionID	Topic	Semester	Relation	ra
0	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	Father	
1	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	Father	
2	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	Father	
3	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	Father	
4	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	Father	

In [109…

```
data.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 [110…

```
data.dtypes
```

Out[110]:
```
gender                     object
NationalITy                object
PlaceofBirth               object
StageID                    object
GradeID                    object
SectionID                  object
Topic                      object
Semester                   object
Relation                   object
raisedhands                 int64
VisITedResources            int64
AnnouncementsView           int64
Discussion                  int64
ParentAnsweringSurvey      object
ParentschoolSatisfaction   object
StudentAbsenceDays         object
Class                      object
dtype: object
```

In [111…
```
data.isnull().sum()
```

Out[111]:
```
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 [112…
```
data.describe(include='object')
```

Out[112]:

	gender	NationalITy	PlaceofBirth	StageID	GradeID	SectionID	Topic	Semester	Re
count	480	480	480	480	480	480	480	480	
unique	2	14	14	3	10	3	12	2	
top	M	KW	KuwaIT	MiddleSchool	G-02	A	IT	F	
freq	305	179	180	248	147	283	95	245	

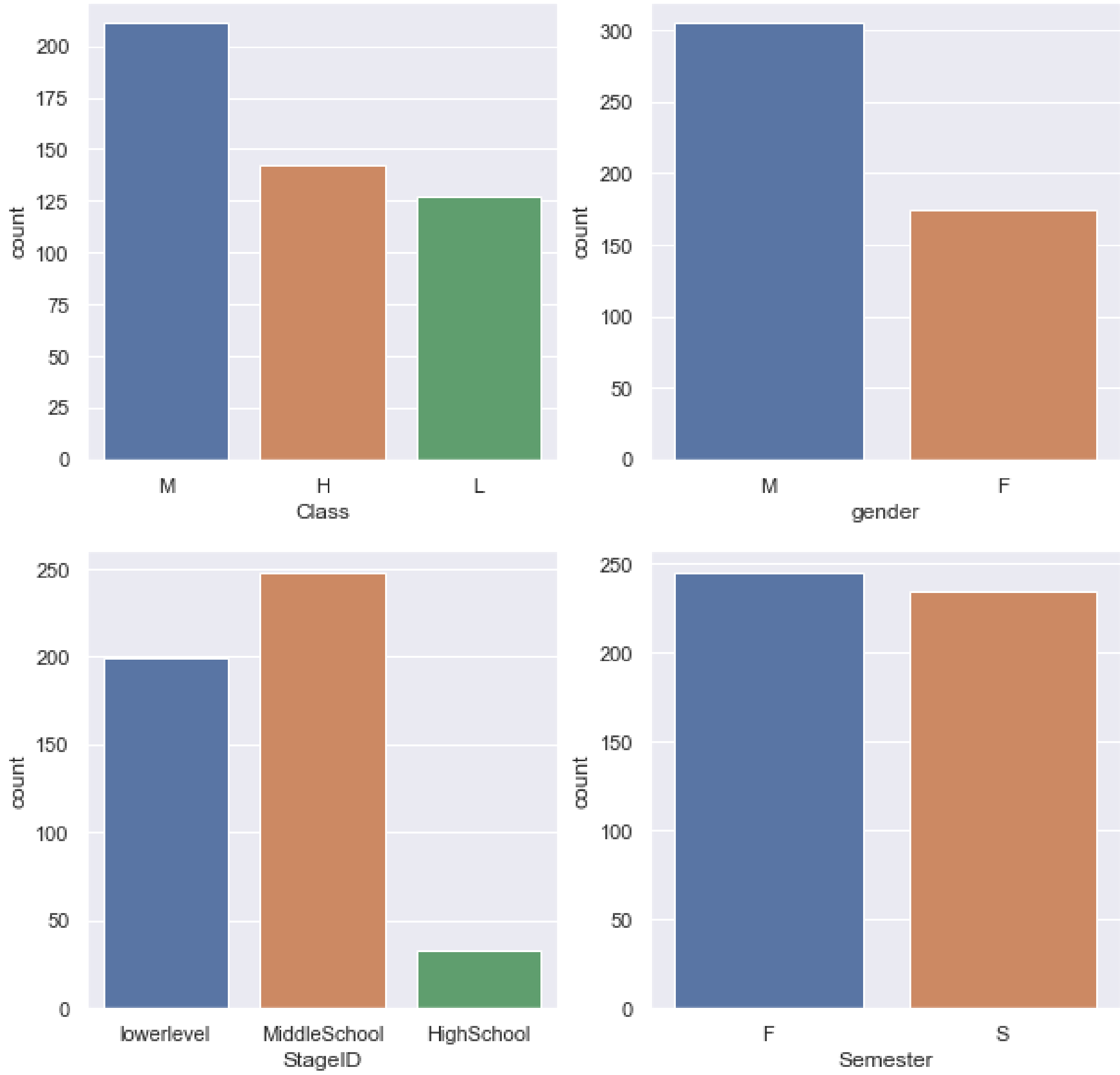
In [113…
```
unique_values = data.apply(lambda col: col.nunique())

# Display the unique values
print(unique_values)
```

```
gender                      2
NationalITy                14
PlaceofBirth               14
StageID                     3
GradeID                    10
SectionID                   3
Topic                      12
Semester                    2
Relation                    2
raisedhands                82
VisITedResources           89
AnnouncementsView          88
Discussion                 90
ParentAnsweringSurvey       2
ParentschoolSatisfaction    2
StudentAbsenceDays          2
Class                       3
dtype: int64
```

1. Visualize just the categorical features individually to see what options are included and how each option fares when it comes to count(how many times it appears) and see what can be deduce from that?

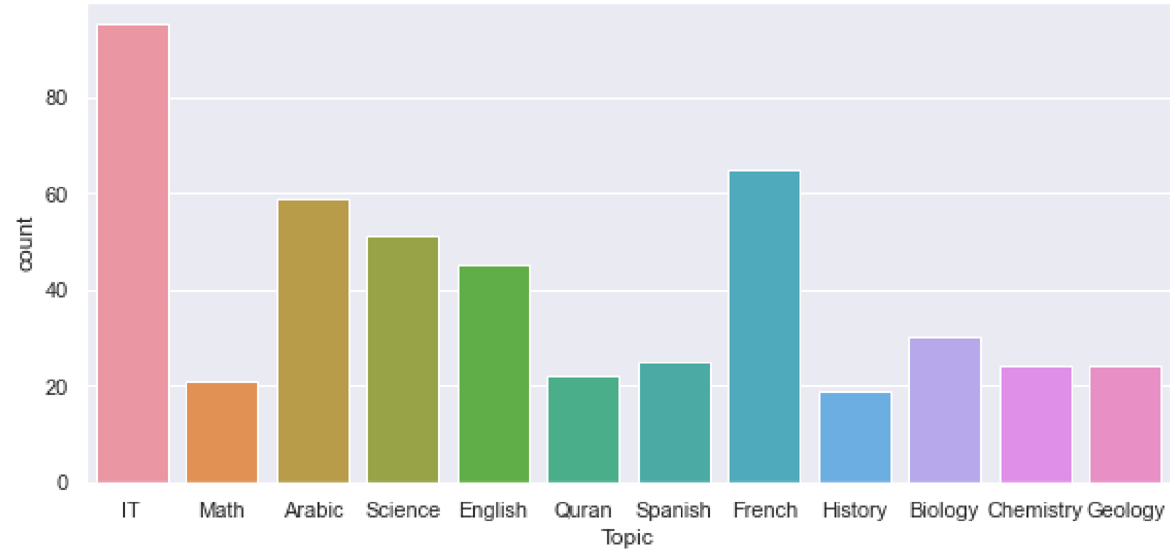
In [114…

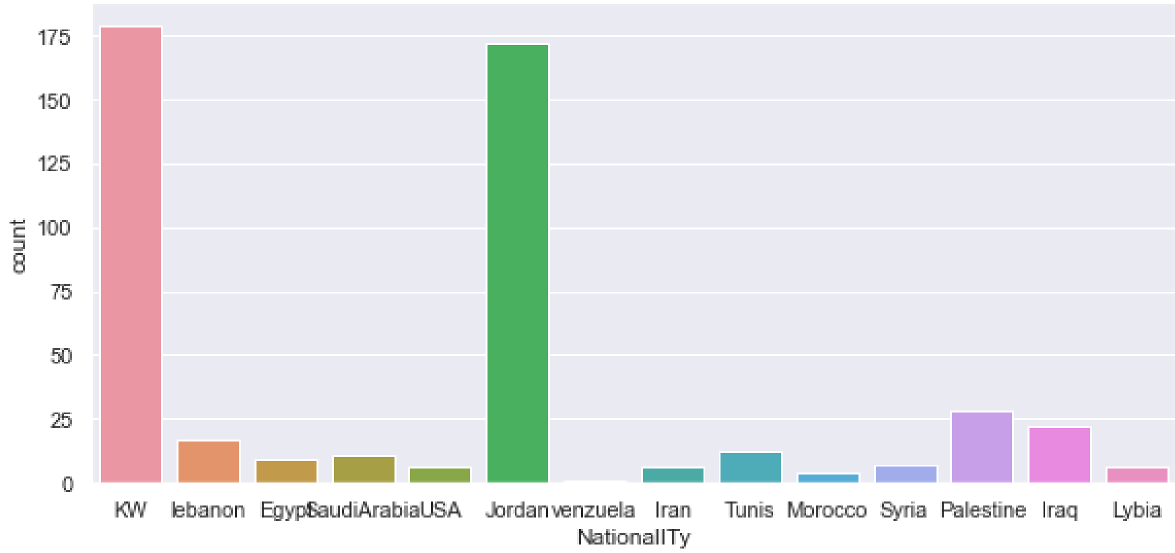
```
plt.figure(figsize=(10,10))
plt.subplot(2,2,1)
sns.countplot(x='Class', data=data, order=['M','H','L'])
plt.subplot(2,2,2)
sns.countplot(x='gender', data=data, order=['M','F'])
plt.subplot(2,2,3)
sns.countplot(x='StageID', data=data)
plt.subplot(2,2,4)
sns.countplot(x='Semester', data=data)
plt.show()
```



In [115…

```
fig, (axis1, axis2)  = plt.subplots(2, 1,figsize=(10,10))
sns.countplot(x='Topic', data=data, ax=axis1)
sns.countplot(x='NationalITy', data=data, ax=axis2)
plt.show()
```

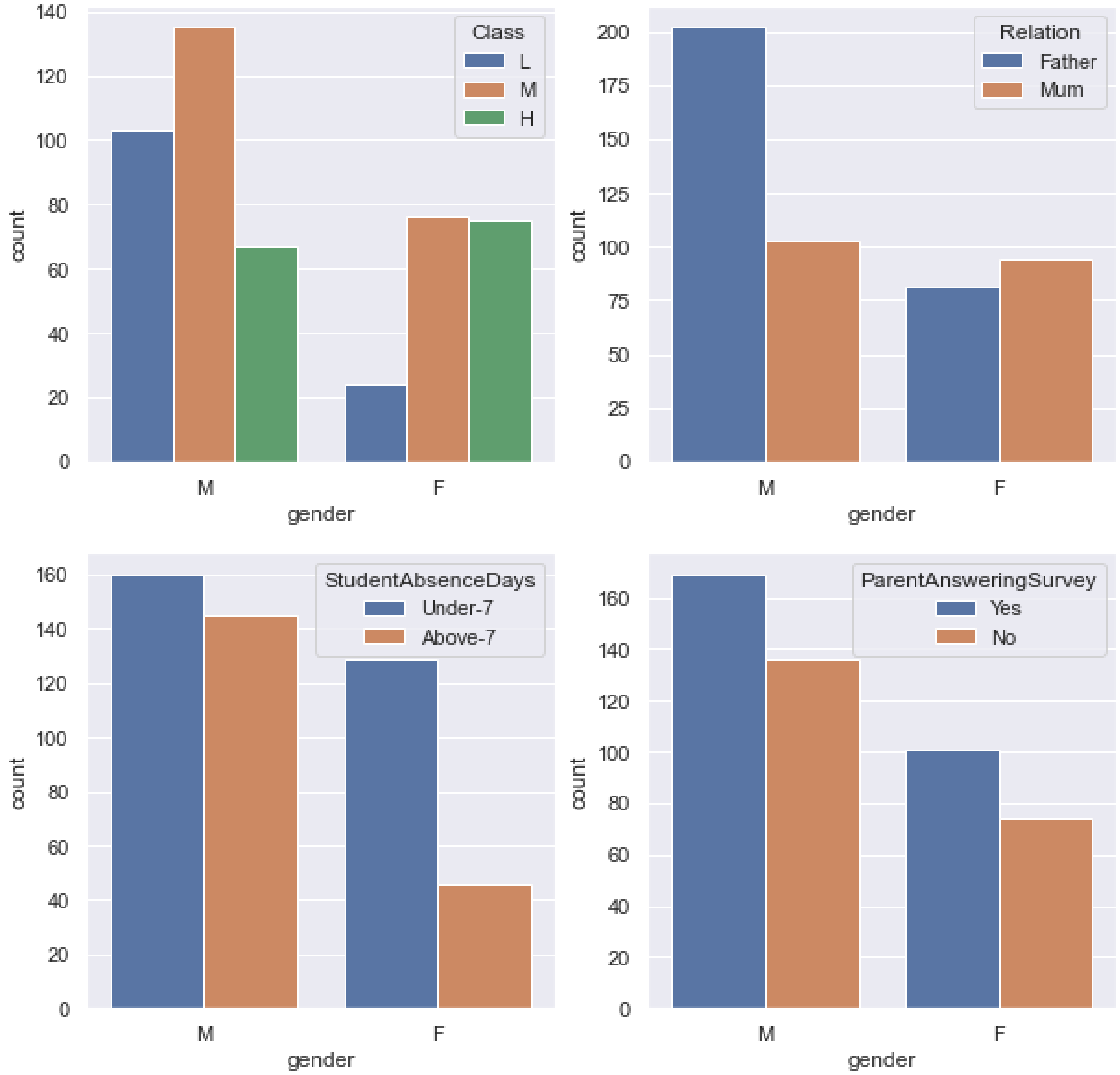




2. Look at some categorical features in relation to each other, to see what insights could be possibly read?

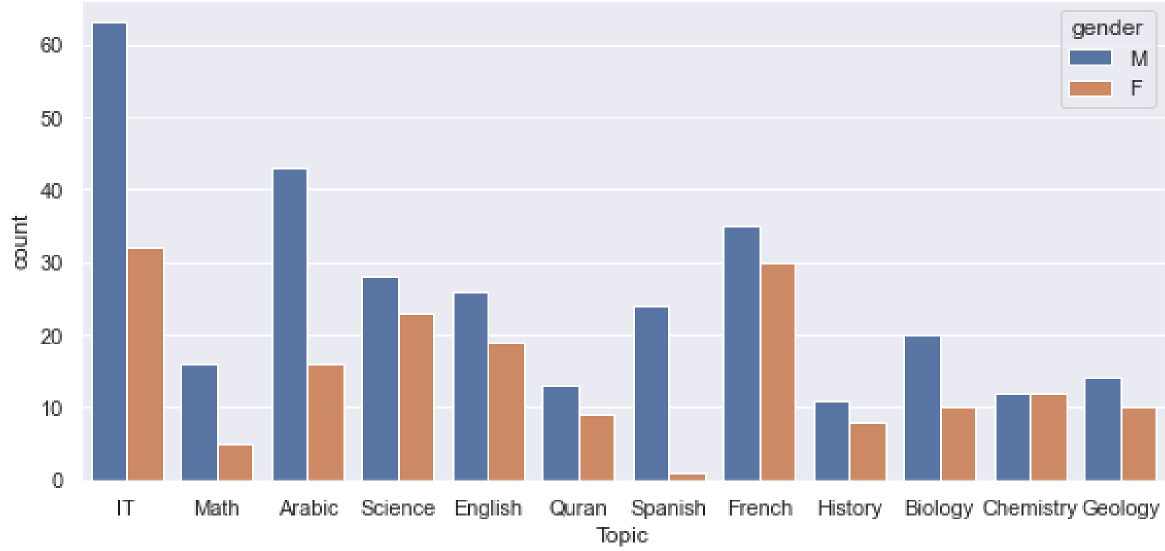
In [116…

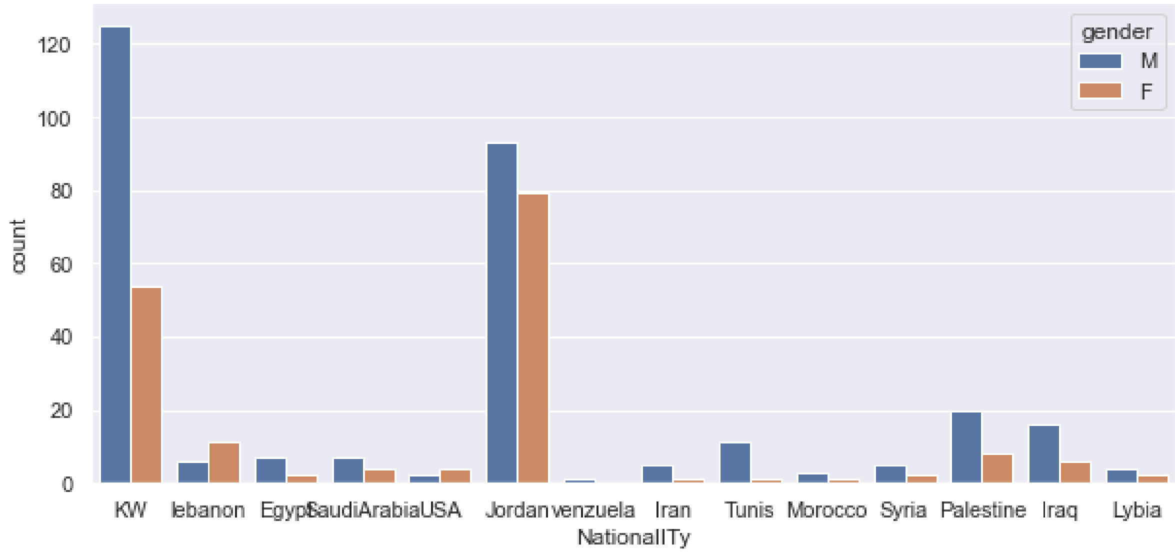
```
fig, axarr  = plt.subplots(2,2,figsize=(10,10))
sns.countplot(x='gender', hue='Class', data=data, ax=axarr[0,0], order=['M','F'], h
sns.countplot(x='gender', hue='Relation', data=data, ax=axarr[0,1], order=['M','F']
sns.countplot(x='gender', hue='StudentAbsenceDays', data=data, ax=axarr[1,0], order
sns.countplot(x='gender', hue='ParentAnsweringSurvey', data=data, ax=axarr[1,1], or
plt.show()
```



In [117…

```
fig, (axis1, axis2)  = plt.subplots(2, 1,figsize=(10,10))
sns.countplot(x='Topic', hue='gender', data=data, ax=axis1)
sns.countplot(x='NationalITy', hue='gender', data=data, ax=axis2)
plt.show()
```



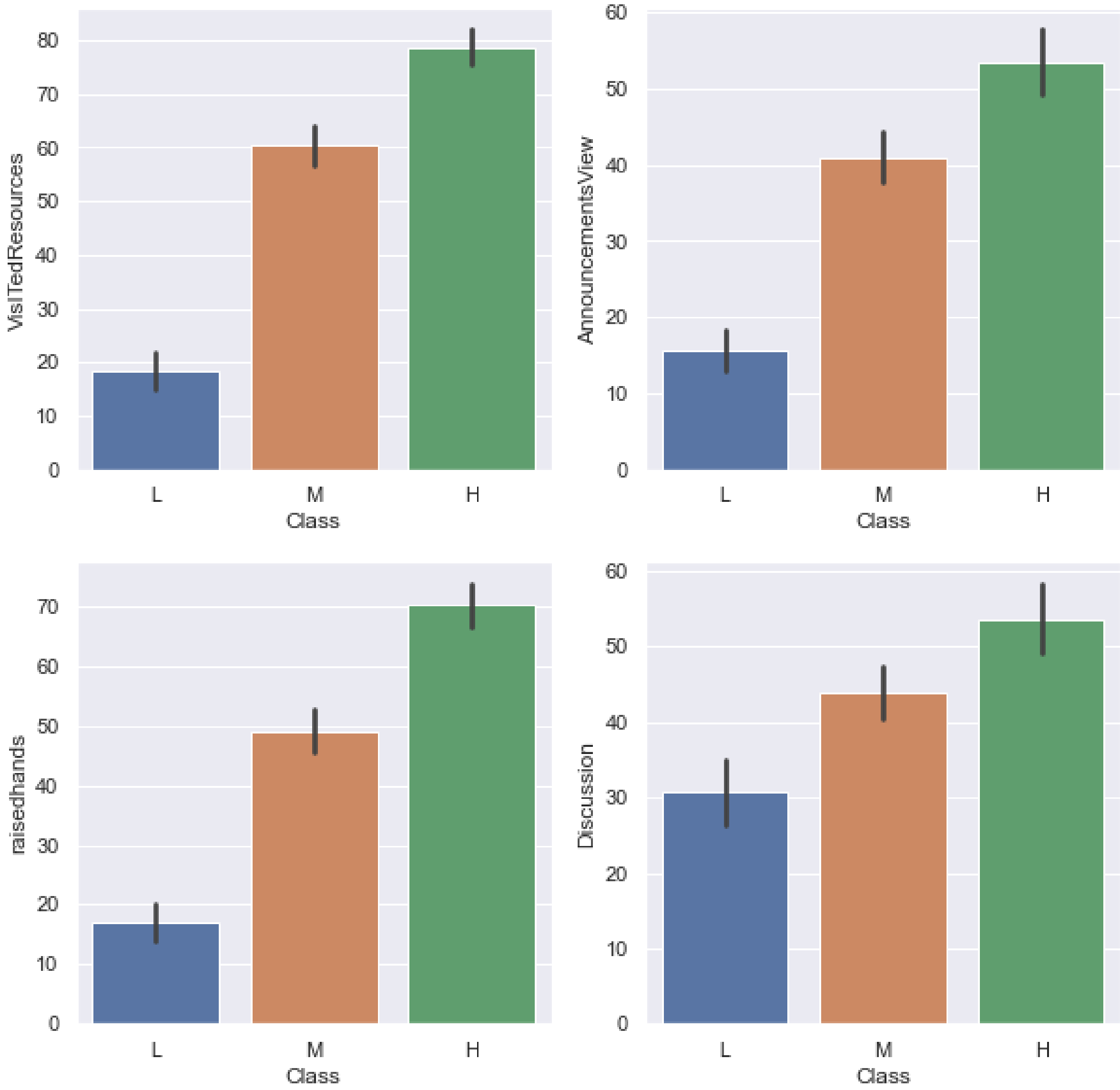


- Girls seem to have performed better than boys
- In the case of girls, mothers seem to be more interested in their education than fathers
- Girls had much better attendance than boys
- No apparent gender bias when it comes to subject/topic choices, we cannot conclude that girls performed better because they perhaps took less technical subjects
- Gender disparity holds even at a country level. May just be as a result of the sampling

3. Visualize categorical variables with numerical variables and give conclusions?

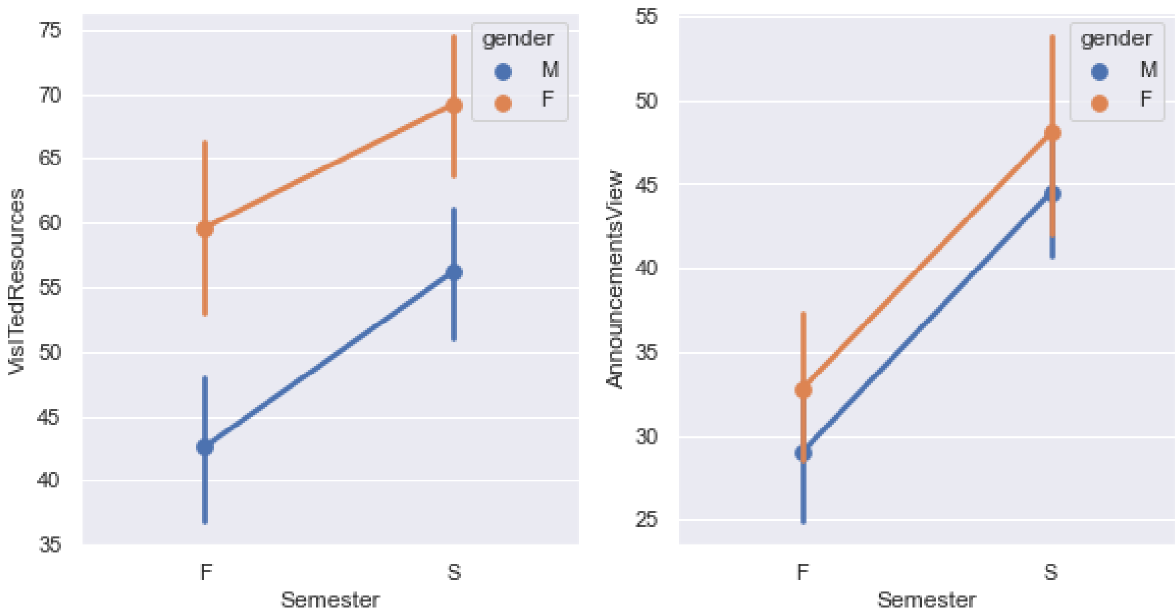
In [118…

```
plt.figure(figsize=(10,10))
plt.subplot(2,2,1)
sns.barplot(x='Class', y='VisITedResources', data=data, order=['L','M','H'])
plt.subplot(2,2,2)
sns.barplot(x='Class', y='AnnouncementsView', data=data, order=['L','M','H'])
plt.subplot(2,2,3)
sns.barplot(x='Class', y='raisedhands', data=data, order=['L','M','H'])
plt.subplot(2,2,4)
sns.barplot(x='Class', y='Discussion', data=data, order=['L','M','H'])
plt.show()
```



In [119…

```
plt.figure(figsize=(10,5))
plt.subplot(1, 2, 1)
sns.pointplot(x='Semester', y='VisITedResources', hue='gender', data=data)
plt.subplot(1, 2, 2)
sns.pointplot(x='Semester', y='AnnouncementsView', hue='gender', data=data)
plt.show()
```



Ans :

- As expected, those that participated more (higher counts in Discussion, raisedhands, AnnouncementViews, RaisedHands), performed better
- In the case of both visiting resources and viewing announcements, students were more vigilant in the second semester, perhaps that last minute need to boost your final grade

In [120…

```
ave_raisedhands = sum(data['raisedhands'])/len(data['raisedhands'])
ave_VisITedResources = sum(data['VisITedResources'])/len(data['VisITedResources'])
ave_AnnouncementsView = sum(data['AnnouncementsView'])/len(data['AnnouncementsView'
unsuccess = data.loc[(data['raisedhands'] >= ave_raisedhands) & (data['VisITedResou
```

In [121…

```
unsuccess
```

Out[121]:

	gender	NationalITy	PlaceofBirth	StageID	GradeID	SectionID	Topic	Semester	Re
444	M	Jordan	Jordan	MiddleSchool	G-08	A	Chemistry	F	
445	M	Jordan	Jordan	MiddleSchool	G-08	A	Chemistry	S	

4. From the above result, what are the factors that leads to get low grades of the students?

Note : Above two students have features ('raisedhands' , 'VisITedResources' , 'AnnouncementsView') greater than average

In [122…

```
data['numeric_class'] = [1 if data.loc[i,'Class'] == 'L' else 2 if data.loc[i,'Clas
```

In [123…

```
grade_male_ave = sum(data[data.gender == 'M'].numeric_class)/float(len(data[data.ge
grade_female_ave = sum(data[data.gender == 'F'].numeric_class)/float(len(data[data.
```

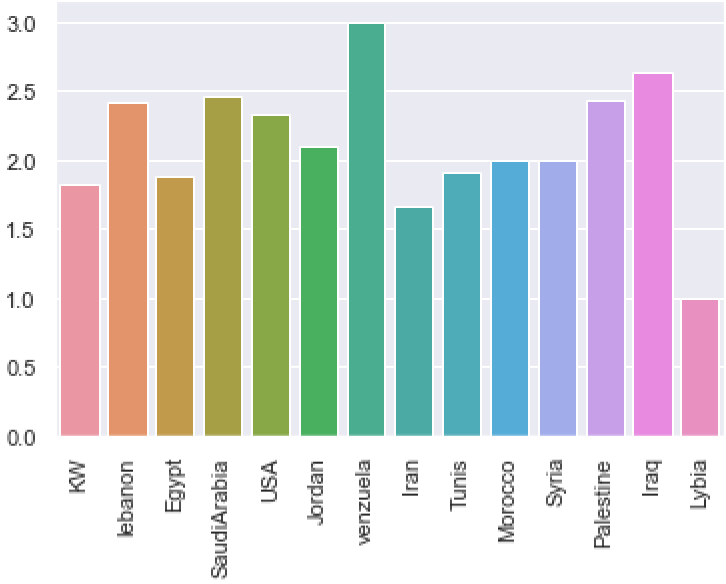
- Gender comparison cannot completely explain low level grades

In [124…

```
# Now lets look at nationality
nation = data.NationalITy.unique()
nation_grades_ave = [sum(data[data.NationalITy == i].numeric_class)/float(len(data[
ax = sns.barplot(x=nation, y=nation_grades_ave)
jordan_ave = sum(data[data.NationalITy == 'Jordan'].numeric_class)/float(len(data[d
print('Jordan average: '+str(jordan_ave))
plt.xticks(rotation=90)
```

```
Jordan average: 2.0930232558139537
```

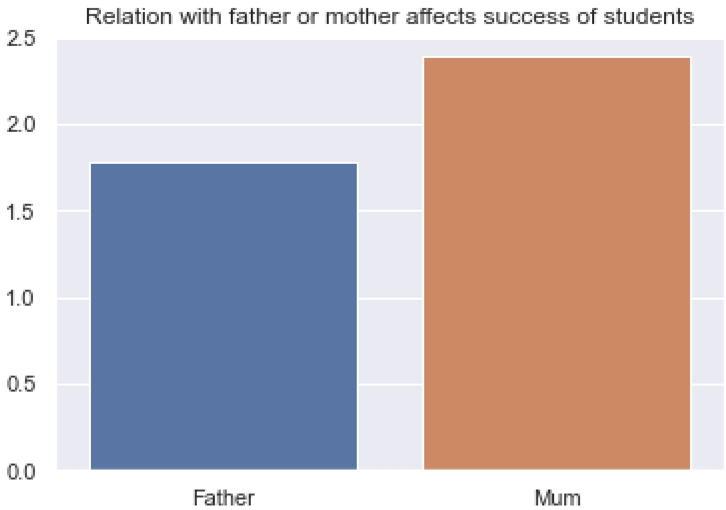
Out[124]:
```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13]),
 [Text(0, 0, 'KW'),
  Text(1, 0, 'lebanon'),
  Text(2, 0, 'Egypt'),
  Text(3, 0, 'SaudiArabia'),
  Text(4, 0, 'USA'),
  Text(5, 0, 'Jordan'),
  Text(6, 0, 'venzuela'),
  Text(7, 0, 'Iran'),
  Text(8, 0, 'Tunis'),
  Text(9, 0, 'Morocco'),
  Text(10, 0, 'Syria'),
  Text(11, 0, 'Palestine'),
  Text(12, 0, 'Iraq'),
  Text(13, 0, 'Lybia')])
```



- Gender comparison cannot completely explain low level grades

In [125…
```
# Lets look at relation with family members
relation = data.Relation.unique()
relation_grade_ave = [sum(data[data.Relation == i].numeric_class)/float(len(data[da
ax = sns.barplot(x=relation, y=relation_grade_ave)
plt.title('Relation with father or mother affects success of students')
```

Out[125]:
```
Text(0.5, 1.0, 'Relation with father or mother affects success of students')
```



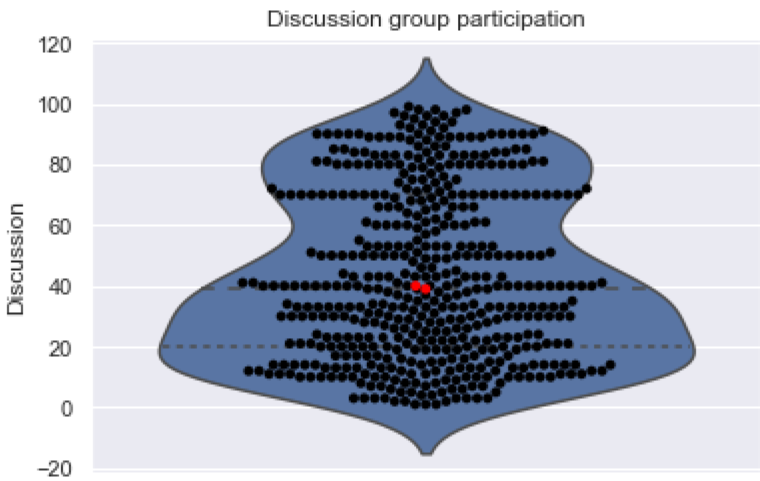
- Having relation with mum has positive effect on these students

In [126…

```
#Lets look at how many times the student participate on discussion groups
discussion = data.Discussion
discussion_ave = sum(discussion)/len(discussion)
ax = sns.violinplot(y=discussion,split=True,inner='quart')
ax = sns.swarmplot(y=discussion,color='black')
ax = sns.swarmplot(y = unsuccess.Discussion, color='red')
plt.title('Discussion group participation')
```

Out[126]:

```
Text(0.5, 1.0, 'Discussion group participation')
```

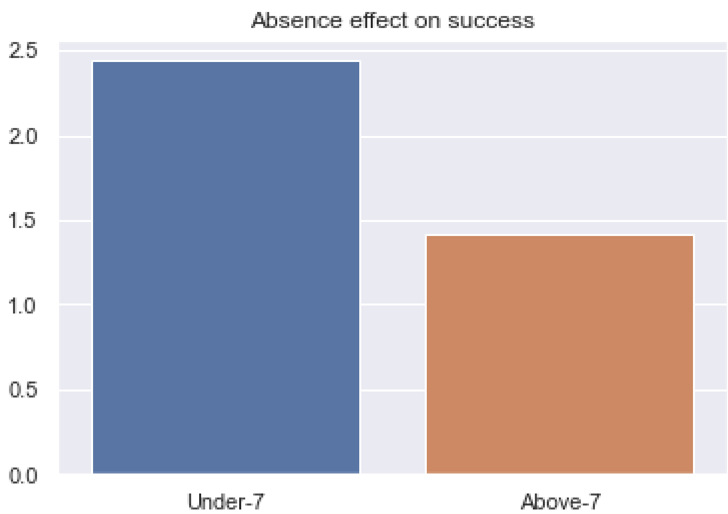


In [127…

```
# Now lastly lets look at
absence_day = data.StudentAbsenceDays.unique()
absense_day_ave = [sum(data[data.StudentAbsenceDays == i].numeric_class)/float(len(
ax = sns.barplot(x=absence_day, y=absense_day_ave)
plt.title('Absence effect on success')
```

Out[127]:

```
Text(0.5, 1.0, 'Absence effect on success')
```



Ans :

- These two students are under the average of discussion (43). Therefore, not participating in discussion groups can be important reason to get low grades
- Their absence days are above seven which resulted in low grades

5. Build classification model and present it's classification report ?

In [128…
```
data.head()
```

Out[128]:

	gender	NationalITy	PlaceofBirth	StageID	GradeID	SectionID	Topic	Semester	Relation	ra
0	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	Father	
1	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	Father	
2	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	Father	
3	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	Father	
4	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	Father	

In [129…
```
data1 = data.drop('Class',axis = 1)
data_with_dummies = pd.get_dummies(data1, drop_first=True)
```

In [130…
```
data_with_dummies.head()
```

Out[130]:

	raisedhands	VisITedResources	AnnouncementsView	Discussion	numeric_class	gender_M	Nati
0	15	16	2	20	2	1	
1	20	20	3	25	2	1	
2	10	7	0	30	1	1	
3	30	25	5	35	1	1	
4	40	50	12	50	2	1	

5 rows × 61 columns

In [131…
```
Features = data_with_dummies.drop(['numeric_class'],axis = 1)
Target = data_with_dummies['numeric_class']
```

In [132…
```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
scaler.fit(Features)
```

Out[132]:
```
StandardScaler()
```

In [133…
```
X = scaler.fit_transform(Features)
```

In [134…
```
X_train, X_test, y_train, y_test = train_test_split(X, Target, test_size=0.3, rando
```

In [135…
```
Logit_Model = LogisticRegression()
Logit_Model.fit(X_train,y_train)
```

Out[135]:
```
LogisticRegression()
```

In [136…
```
Prediction = Logit_Model.predict(X_test)
Score = accuracy_score(y_test,Prediction)
Report = classification_report(y_test,Prediction)
```

In [137…
```
Prediction
```

Out[137]:
```
array([2, 2, 3, 1, 1, 1, 1, 3, 2, 2, 2, 3, 2, 2, 1, 1, 1, 2, 1, 1, 3, 3,
       2, 3, 2, 2, 3, 2, 2, 3, 3, 3, 3, 2, 2, 2, 3, 2, 2, 3, 1, 3, 2, 1,
       2, 2, 3, 2, 2, 2, 2, 1, 2, 2, 2, 2, 3, 2, 3, 1, 3, 1, 2, 2, 2, 2,
       1, 2, 1, 2, 2, 2, 1, 2, 2, 1, 2, 3, 2, 1, 2, 2, 3, 2, 3, 3, 3, 3,
       2, 3, 2, 1, 2, 1, 3, 3, 2, 3, 2, 3, 2, 1, 2, 1, 2, 2, 3, 2, 2, 1,
       3, 2, 2, 3, 2, 2, 2, 2, 1, 1, 3, 1, 3, 1, 3, 3, 1, 3, 3, 3, 1, 3,
       3, 3, 2, 1, 1, 1, 3, 2, 2, 1, 2, 2], dtype=int64)
```

In [138…
```
Score
```

Out[138]:
```
0.7361111111111112
```

In [139…
```
print(Report)
```

```
              precision    recall  f1-score   support

           1       0.76      0.87      0.81        30
           2       0.78      0.70      0.74        74
           3       0.65      0.70      0.67        40

    accuracy                           0.74       144
   macro avg       0.73      0.76      0.74       144
weighted avg       0.74      0.74      0.74       144
```

In [ ]:
```

```