Introduction to Big Data Analytics

Term Project Report

Topic: 學生學業成績分析

張育祿 二資陸生二 107AEA002
李子健 二資陸生二 108AEA001
王翔 二資陸生二 108AEA008

December 2020

動機：

學業成績對於一個學生來說是非常重要的，但我們並不知道什麼是影響成績最大的因素。我們會通過對資料集的分析，建立模型來預測學生成績。

計畫摘要：

我們利用資料中的受教育程度，班級，選擇課程，成績，出勤特徵，以及家長參與等信息，來進行資料分析及視覺化，通過分析資料並建立模型預測學生成績。

研究步驟：

資料预处理->資料視覺化->模型建立分析->参数调优->预测效果

環境：

Python 3.7 + Jupyter Notebook

Python 所需套件：

pandas、sklearn、seaborn、matplotlib、numpy

參考資料：

Students' Academic Performance Dataset (xAPI-Edu-Data)
https://www.kaggle.com/aljarah/xAPI-Edu-Data
Seaborn API Website
https://seaborn.pydata.org/api.html

資料集栏位介紹：

1. gender-学生性别(“M”或“FM”)
2. National-學生國籍(’ Kuwait’,’ Lebanon’,’ Egypt’,’ SaudiArabia’,’ USA’,’ Jordan’,’ Venezuela’,’ Iran’,’ Tunis’,’ Morocco’,’ Syria’,’ Palestine’,’ Iraq’,’ Lybia)
3. PlaceofBirth-學生出生地(“KuwaIT”、“Jordan”、“Iraq ” 、“lebanon”、“SaudiArabia ”、“USA ”、“Palestine”、“Egypt”、“Tunis”、“Iran”、“Lybia ”、“Syria ”、“Morocco”、“venzuela”)
4. StageID-學生所屬教育級別(“lowerlevel”、“MiddleSchool ”、“HighSchool”)
5. GradeID-年級(“G-01”、“G-02”、“G-03”、“G-04”、“G-05”、“G-06”、“G-07”、“G-08”、“G-09”、“G-10”、“G-11”、“G-12”)
6. SectionID-學生所屬教室(‘A’，‘B’，‘C’)
7. Topic—課程('IT'、 'Math'、 'Arabic' 、'Science'、'English'、 'Quran' 、'Spanish' 、'French' 、'History'、 'Biology' 'Chemistry' 、'Geology')
8. Semester-學年(“F”、“S”)
9. Relation-家長與學生之關係(’mom’,’father’)

10. raisedhands-學生在課堂上有舉手的次數(数字：0-100)
11. VisITedResources-學生訪問課程內容的次數(数字：0-100)
12. AnnouncementsView-學生檢查新公告的次數(数字：0-100)
13. Discussion-學生參與討論小組的次數(数字：0-100)
14. ParentAnsweringSurvey-家長是否回答學校提供的調查('Yes','No')
15. ParentschoolSatisfaction-家長對學校的滿意度('Yes','No')
16. StudentAbsenceDays-每名學生缺席天數(above-7, under-7)
17. Class-學生成績分類(L、M、H)

分析過程：

1. 載入套件及資料

```
1. import pandas as pd
2. import numpy as np
3. import seaborn as sns
4. import matplotlib.pyplot as plt
5.
6. from sklearn import preprocessing, svm
7. from sklearn.linear_model import Perceptron
8. from sklearn.tree import DecisionTreeClassifier
9.
10.data = pd.read_csv('xAPI-Edu-Data.csv')
11.data.head()
```

Out[1]:

	gender	NationalITy	PlaceofBirth	StageID	GradeID	SectionID	Topic	Semester	Relation	raisedhands	VisITedResources	AnnouncementsView	Discussior
0	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	Father	15	16	2	2(
1	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	Father	20	20	3	2!
2	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	Father	10	7	0	3(
3	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	Father	30	25	5	3!
4	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	Father	40	50	12	5(

查看資料

2. 資料預處理

2.1 查看資料規格

Input: data.info()

Output:

```
1. <class 'pandas.core.frame.DataFrame'>
2. RangeIndex: 480 entries, 0 to 479
3. Data columns (total 17 columns):
4.  #   Column                    Non-Null Count  Dtype
5. ---  ------                    --------------  -----
```

```
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
```

根據上表分析可知，不存在空值，資料不需要進行預處理。

2.2 查看學生成績類別

Input: data.Class.unique()

Output: array(['M', 'L', 'H'], dtype=object)

學生成績一共分為三類['L', 'M', 'H']，這將作為評判學生的標準。

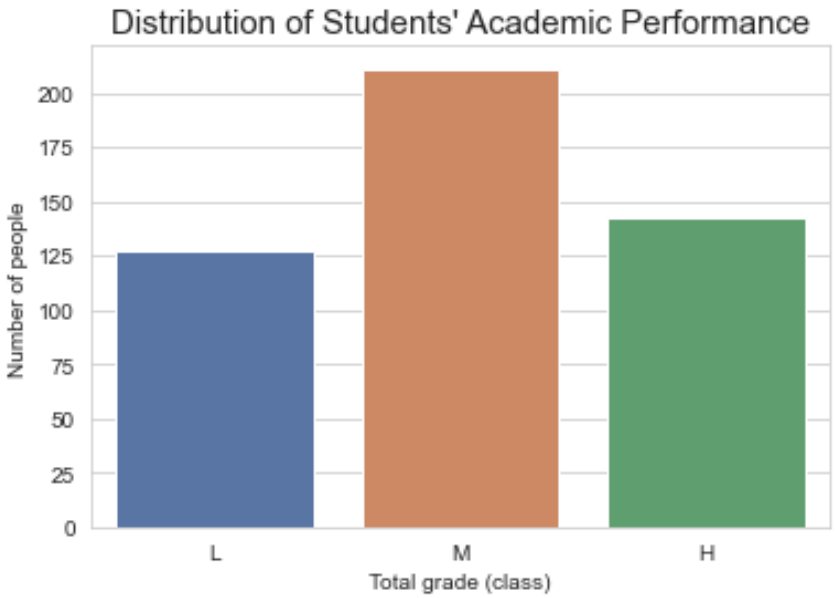
L:0-59 不及格；

M:60-89 中等；

H:90-100 高分。

3. 資料視覺化

```
sns.set_style("whitegrid")
ax = sns.countplot(x='Class', data=data, order=['L', 'M', 'H'], palette="deep")
plt.xlabel('Total grade (class)')
plt.ylabel('Number of people')
plt.title("Distribution of Students' Academic Performance", size=15)
plt.show()
```



根據上圖得知，大部分學生都處於中等成績，高分其次，不及格的人數最少。

3.1 查看不及格學生的信息

```
1. data.loc[data["Class"]=='L']
```

	gender	NationalITy	PlaceofBirth	StageID	GradeID	SectionID	Topic	Semester	Relation	raisedhands	VisITedResources	AnnouncementsView
2	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	Father	10	7	0
3	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	Father	30	25	5
6	M	KW	KuwaIT	MiddleSchool	G-07	A	Math	F	Father	35	12	0
12	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	Father	5	1	0
13	M	lebanon	lebanon	MiddleSchool	G-08	A	Math	F	Father	20	14	12
...	...	...	...	...	...	...	...	...	...	...	...	...
469	F	Jordan	Jordan	MiddleSchool	G-08	A	Chemistry	S	Father	9	6	15
474	F	Jordan	Jordan	MiddleSchool	G-08	A	Chemistry	F	Father	2	7	4
475	F	Jordan	Jordan	MiddleSchool	G-08	A	Chemistry	S	Father	5	4	5
478	F	Jordan	Jordan	MiddleSchool	G-08	A	History	F	Father	30	17	14
479	F	Jordan	Jordan	MiddleSchool	G-08	A	History	S	Father	35	14	23

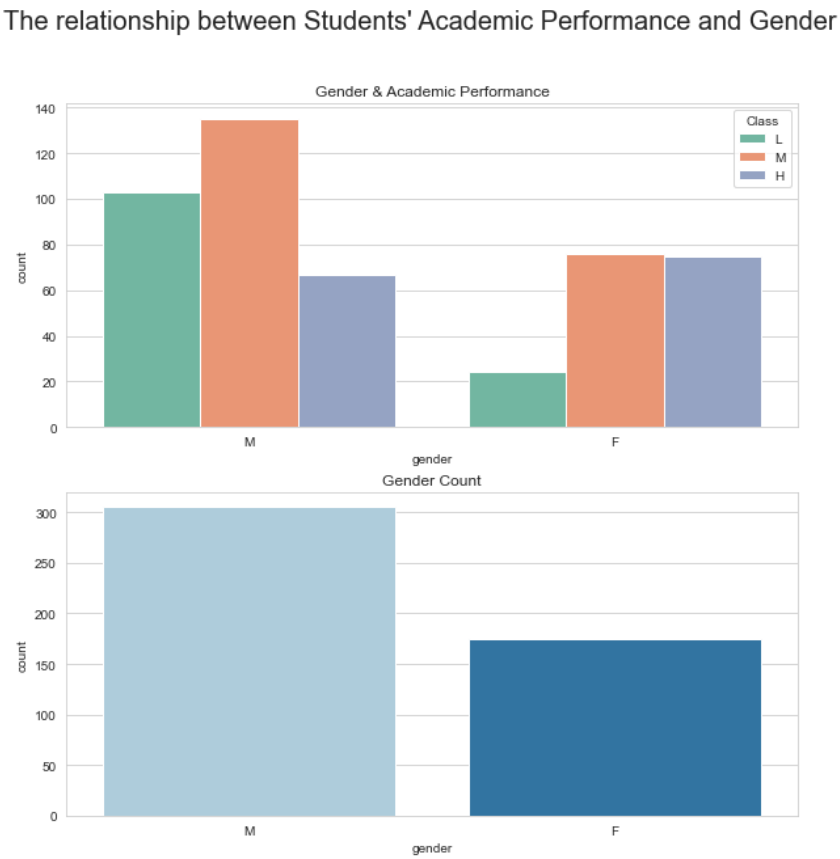
mester	Relation	raisedhands	VisITedResources	AnnouncementsView	Discussion	ParentAnsweringSurvey	ParentschoolSatisfaction	StudentAbsenceDays	Class
F	Father	10	7	0	30	No	Bad	Above-7	L
F	Father	30	25	5	35	No	Bad	Above-7	L
F	Father	35	12	0	17	No	Bad	Above-7	L
F	Father	5	1	0	11	No	Bad	Above-7	L
F	Father	20	14	12	19	No	Bad	Above-7	L
...	...	...	...	...	...	...	...	...	...
S	Father	9	6	15	85	No	Bad	Above-7	L
F	Father	2	7	4	8	No	Bad	Above-7	L
S	Father	5	4	5	8	No	Bad	Above-7	L
F	Father	30	17	14	57	No	Bad	Above-7	L
S	Father	35	14	23	62	No	Bad	Above-7	L

根據以上表格分析得知，似乎不及格的學生缺課天數都超過七天，各項數值都集中在一個很低的區域，例如舉手少的人，一般也不會參與討論（由左下圖得知）。

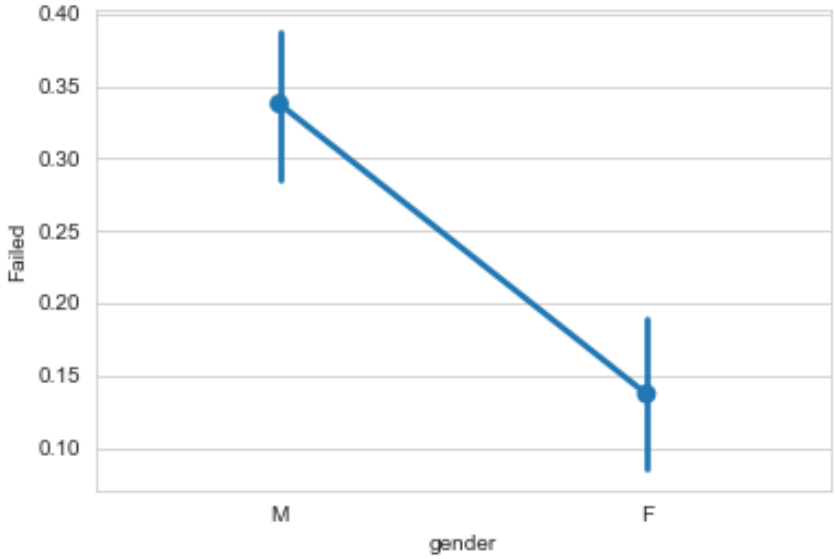
我們還能觀察到不及格的學生一般監護人是父親，且他們沒有接受學校調查、對學校的滿意度不高。

3.2 學生成績與性別的關係

```
1. fig, axarr  = plt.subplots(2,figsize=(10,10))
2. sns.countplot(x='gender', hue='Class', data=data, order=['M', 'F'],hue_order = ['L',
    'M', 'H'], ax=axarr[0], palette="Set2")
3. sns.countplot(x='gender', data=data, order=['M','F'], ax=axarr[1], palette="Paired")

4. axarr[0].set_title('Gender & Academic Performance')
5. axarr[1].set_title('Gender Count')
6. fig.suptitle("The relationship between Students' Academic Performance and Gender", s
   ize=20)
7. plt.show()
```




```
1. sns.pointplot(x='gender', y='Failed', data=data)
```

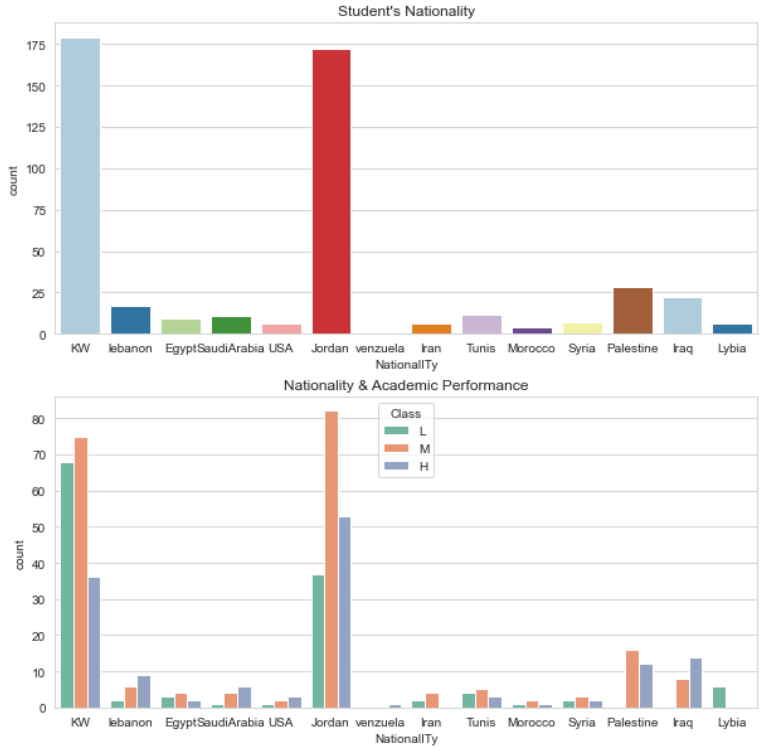


根據此圖分析，我們可以明顯看出女學生的不及格人數要遠遠少於男學生，且女學生的中等和高分段學生人數基本持平，男女學生在高分段人數相差不大。
由此我們可以推測：性別可能影響學生的成績。

3.3 學生成績與國籍的關係

```
1.  fig, axarr  = plt.subplots(2,figsize=(10,10))
2.  axarr[0].set_title("Student's Nationality")
3.  axarr[1].set_title('Nationality & Academic Performance')
4.  fig.suptitle("The relationship between Students' Academic Performance and Nationali
    ty", size=20)
5.  sns.countplot(x='NationalITy', data=data, ax=axarr[0], palette="Paired")
6.  sns.countplot(x='NationalITy', hue='Class', data=data,hue_order = ['L', 'M', 'H'],
    ax=axarr[1], palette="Set2")
7.  plt.show()
```

The relationship between Students' Academic Performance and Nationality



根據此圖分析，除了 KW 和 Jordan 這兩個國籍的學生之外，其餘國籍的學生可分析的樣本數量都比較少，並不能因此推斷出任何有效的信息。

我們能得出的信息有：相比於 KW 的學生，Jordan 學生不及格人數為 KW 學生的一半，總體成績更好一些，而 Iran 和 Lybia 國籍的學生沒有取得高分的。

我們來查看 Iran 和 Lybia 國籍的學生樣本，嘗試分析影響他們學業成績的因素。

查看 Iran 國籍的學生樣本：

```
1. data.loc[data['NationalITy'] == 'Iran']
```

	gender	NationalITy	PlaceofBirth	StageID	GradeID	SectionID	Topic	Semester	Relation	raisedhands	VisITedResources	AnnouncementsView	Di
76	M	Iran	Iran	HighSchool	G-09	A	IT	F	Mum	15	70	37	
126	F	Iran	Iran	lowerlevel	G-02	C	IT	F	Father	2	9	7	
172	M	Iran	Iran	lowerlevel	G-02	B	French	S	Mum	20	22	53	
175	M	Iran	Iran	lowerlevel	G-02	B	French	S	Father	10	2	13	
216	M	Iran	Iran	MiddleSchool	G-08	C	Spanish	S	Mum	27	41	32	
230	M	Iran	Iran	MiddleSchool	G-08	A	Spanish	S	Mum	51	42	12	

查看 Lybia 國籍的學生的樣本：

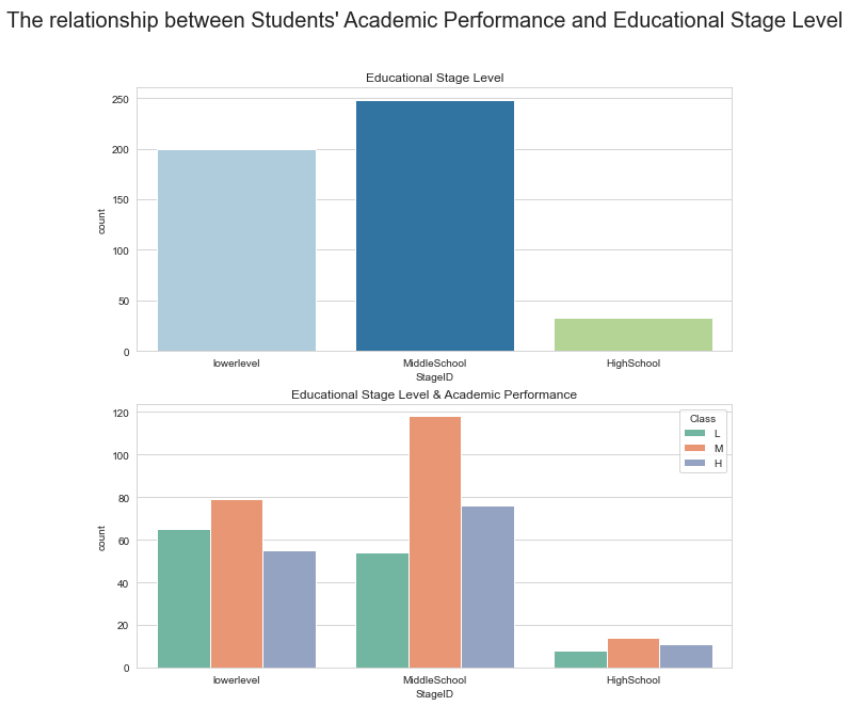
```
1. data.loc[data['NationalITy'] == 'Lybia']
```

	gender	NationalITy	PlaceofBirth	StageID	GradeID	SectionID	Topic	Semester	Relation	raisedhands	VisITedResources	AnnouncementsView	Dis
334	M	Lybia	Lybia	lowerlevel	G-02	A	French	F	Mum	10	8	9	
335	M	Lybia	Lybia	lowerlevel	G-02	A	French	S	Mum	15	7	12	
348	M	Lybia	Lybia	lowerlevel	G-02	B	French	F	Mum	20	3	9	
349	M	Lybia	Lybia	lowerlevel	G-02	B	French	S	Mum	15	4	12	
414	F	Lybia	Lybia	MiddleSchool	G-07	B	Biology	F	Mum	10	9	2	
415	F	Lybia	Lybia	MiddleSchool	G-07	B	Biology	S	Mum	9	7	9	

通過觀察，我們可以得知 Lybia 國籍的學生似乎與所有未通過考試的學生的數據有高度的重合(缺課超過 7 天，數值偏低，沒有學校調查等)。

3.4 學生所屬教育級別與成績的關係

```
1. fig, axarr  = plt.subplots(2,figsize=(10,10))
2. axarr[0].set_title('Educational Stage Level')
3. axarr[1].set_title('Educational Stage Level & Academic Performance')
4. fig.suptitle("The relationship between Students' Academic Performance and Education
   al Stage Level", size=20)
5. sns.countplot(x='StageID', data=data, ax=axarr[0], palette="Paired")
6. sns.countplot(x='StageID', hue='Class', data=data, hue_order = ['L', 'M', 'H'], ax=
   axarr[1], palette="Set2")
7. plt.show()
```

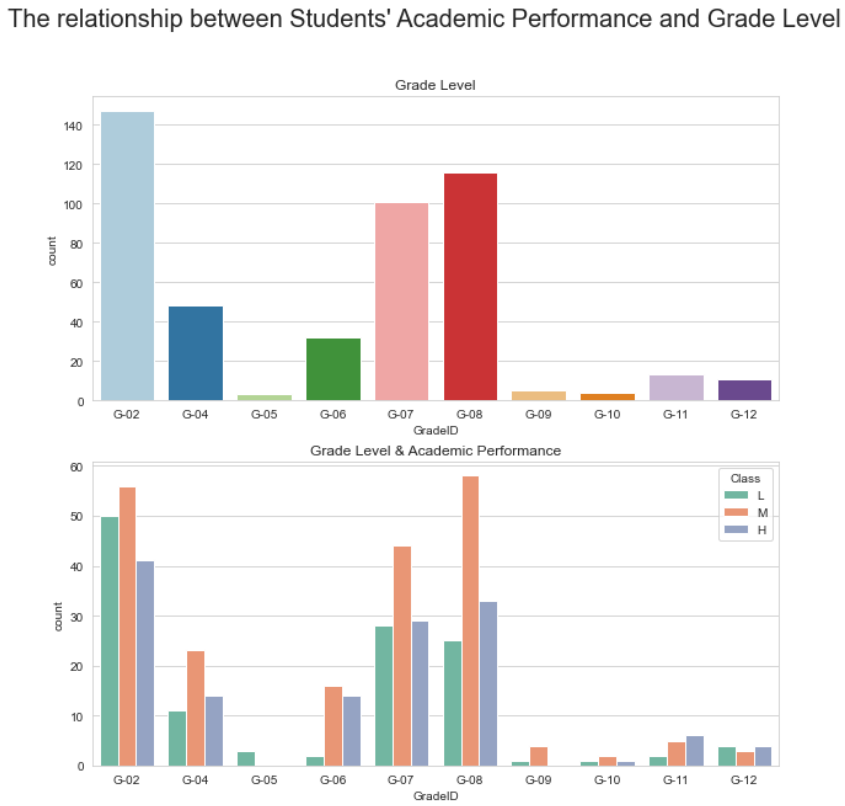


根據此圖分析，我們可以得知不管學生處於哪個教育級別，都是中等成績的人數偏多。

3.5 學生所屬年級與成績的關係

```
1. fig, axarr  = plt.subplots(2,figsize=(10,10))
2. axarr[0].set_title('Grade Level')
```

```
3. axarr[1].set_title('Grade Level & Academic Performance')
4. fig.suptitle("The relationship between Students' Academic Performance and Grade Level", size=20)
5. sns.countplot(x='GradeID',
6.                data=data,
7.                order=['G-02', 'G-04', 'G-05', 'G-06', 'G-07', 'G-08', 'G-09', 'G-10', 'G-11', 'G-12'],
8.                ax=axarr[0], palette="Paired")
9. sns.countplot(x='GradeID',
10.               hue='Class',
11.               data=data,
12.               order=['G-02', 'G-04', 'G-05', 'G-06', 'G-07', 'G-08', 'G-09', 'G-10', 'G-11', 'G-12'],
13.               hue_order = ['L', 'M', 'H'],
14.               ax=axarr[1], palette="Set2")
15.plt.show()
```



根據此圖分析，我們可以得知五年級、九年級、十年級的學生人數很少。除此之外，沒有五年級學生及格，也沒有九年級學生取得高分。

查看五年級學生的樣本：

```
1. data.loc[data['GradeID']=='G-05']
```

	gender	NationalITy	PlaceofBirth	StageID	GradeID	SectionID	Topic	Semester	Relation	raisedhands	VisITedResources	AnnouncementsView	Discu
33	M	KW	KuwaIT	lowerlevel	G-05	A	English	F	Father	8	22	9	
46	M	KW	KuwaIT	lowerlevel	G-05	A	English	F	Father	7	10	1	
60	F	Jordan	Jordan	lowerlevel	G-05	A	English	F	Mum	21	10	28	

查看九年級學生的樣本：

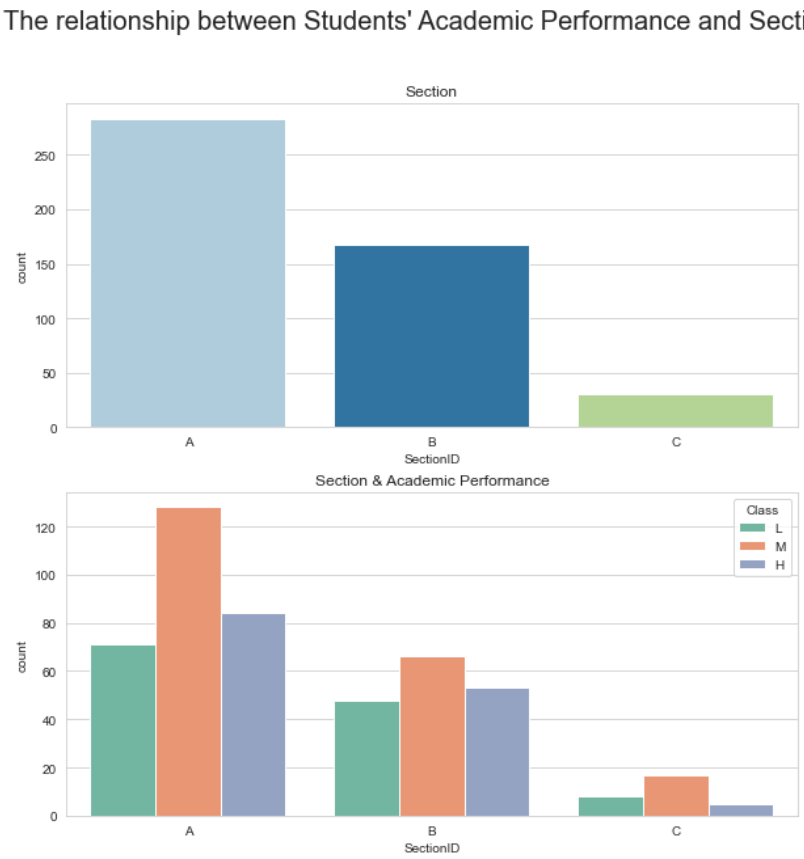
```
1. data.loc[data['GradeID']=='G-09']
```

	gender	NationalITy	PlaceofBirth	StageID	GradeID	SectionID	Topic	Semester	Relation	raisedhands	VisITedResources	AnnouncementsView	Discuss
42	M	KW	KuwaIT	HighSchool	G-09	A	IT	F	Father	10	12	7	
43	F	KW	KuwaIT	HighSchool	G-09	A	IT	F	Father	30	35	28	
44	F	KW	KuwaIT	HighSchool	G-09	A	IT	F	Father	33	33	30	
76	M	Iran	Iran	HighSchool	G-09	A	IT	F	Mum	15	70	37	
77	M	KW	KuwaIT	HighSchool	G-09	A	IT	F	Father	20	80	33	

在觀察後，發現五年級和九年級學生似乎與所有未通過考試的學生的數據有高度的重合(缺課超過 7 天，數值偏低，沒有學校調查等) 。

3.6 學生所屬教室與成績的關係

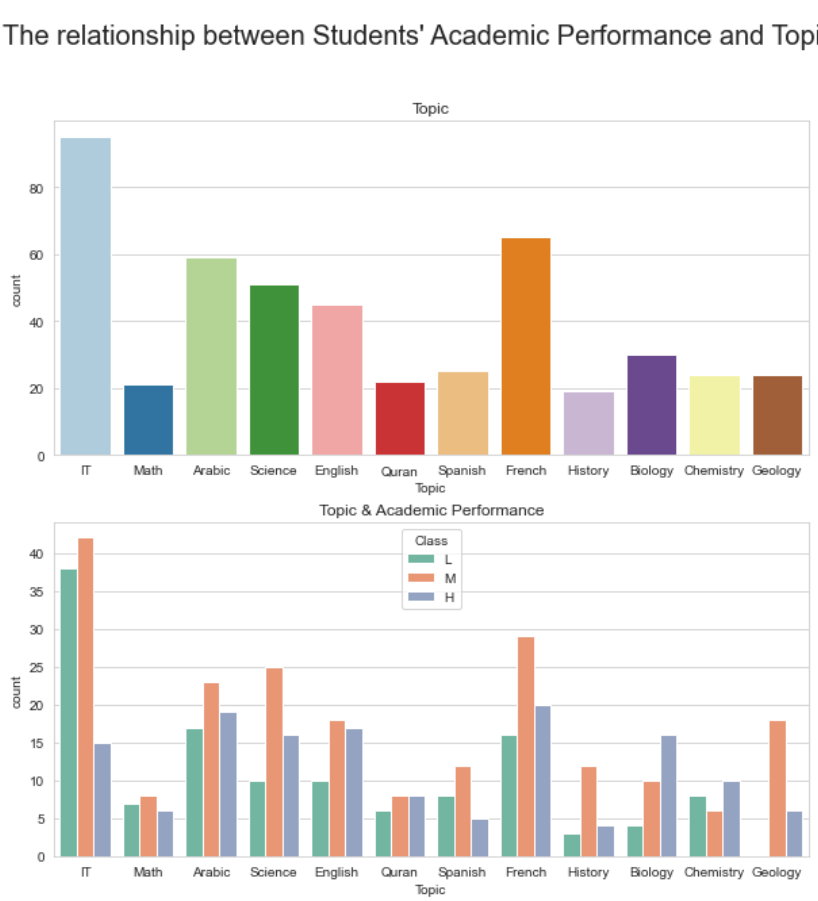
```
1. fig, axarr  = plt.subplots(2,figsize=(10,10))
2. axarr[0].set_title('Section')
3. axarr[1].set_title('Section & Academic Performance')
4. fig.suptitle("The relationship between Students' Academic Performance and Section",
   size=20)
5. sns.countplot(x='SectionID', data=data,
6.                order=['A', 'B', 'C'], ax = axarr[0], palette="Paired")
7. sns.countplot(x='SectionID', hue='Class',
8.                data=data, order=['A', 'B', 'C'],
9.                hue_order = ['L', 'M', 'H'], ax = axarr[1], palette="Set2")
10. plt.show()
```



根據此圖分析，我們得知三個班的總體趨勢都差不多，我們並不能得出什麼有效信息。

3.7 學生所選課程與成績的關係

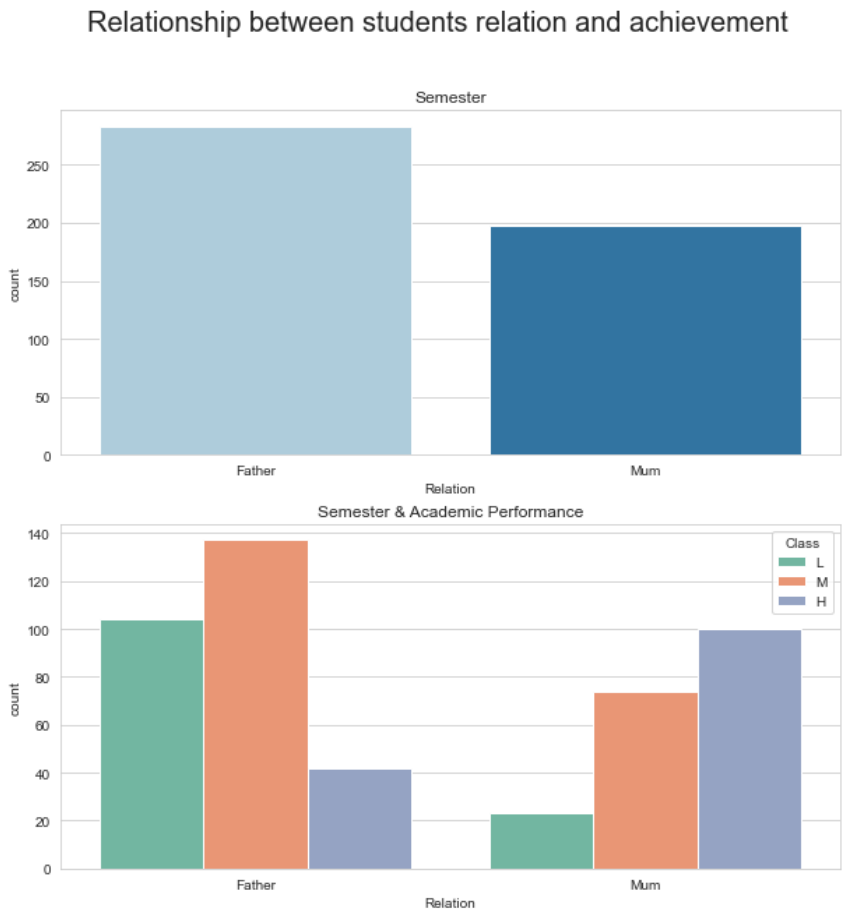
```
1.  fig, axarr  = plt.subplots(2,figsize=(10,10))
2.  axarr[0].set_title('Topic')
3.  axarr[1].set_title('Topic & Academic Performance')
4.  fig.suptitle("The relationship between Students' Academic Performance and Topic", s
    ize=20)
5.  sns.countplot(x='Topic', data=data, ax = axarr[0], palette="Paired")
6.  sns.countplot(x='Topic', hue='Class', data=data,hue_order = ['L', 'M', 'H'], ax = a
    xarr[1], palette="Set2")
7.  plt.show()
```



根據此圖分析，我們可以看到一個有趣的現象，Geology 課程沒有不及格的學生。這是為什麼呢？

3.8 不同學年學生與成績的關係

```
1.  fig, axarr  = plt.subplots(2,figsize=(10,10))
2.  axarr[0].set_title('Semester')
3.  axarr[1].set_title('Semester & Academic Performance')
4.  fig.suptitle("The relationship between Students' Academic Performance and Semester"
    , size=20)
5.  sns.countplot(x='Semester', data=data, ax = axarr[0], palette="Paired")
6.  sns.countplot(x='Semester', hue='Class', data=data,hue_order = ['L', 'M', 'H'], ax
    = axarr[1], palette="Set2")
7.  plt.show()
```

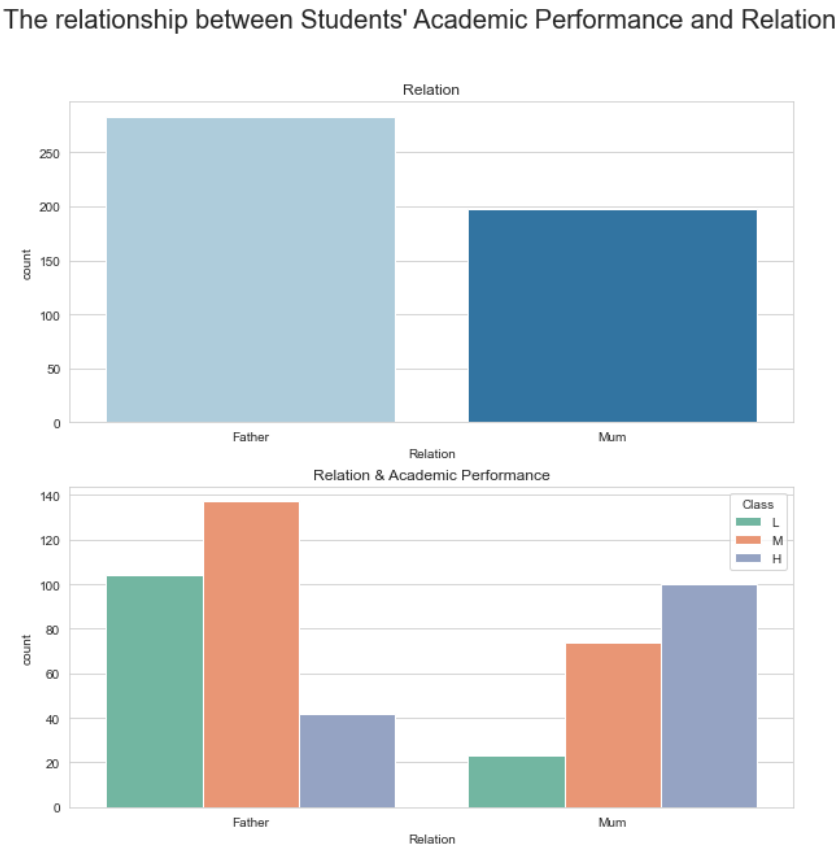


根據此圖分析，第二學年的不及格人數比第一學年少，高分人數都比第一學年多。

由此我們可以推測：學年可能會影響成績。

3.9 學生監護人與成績的關係

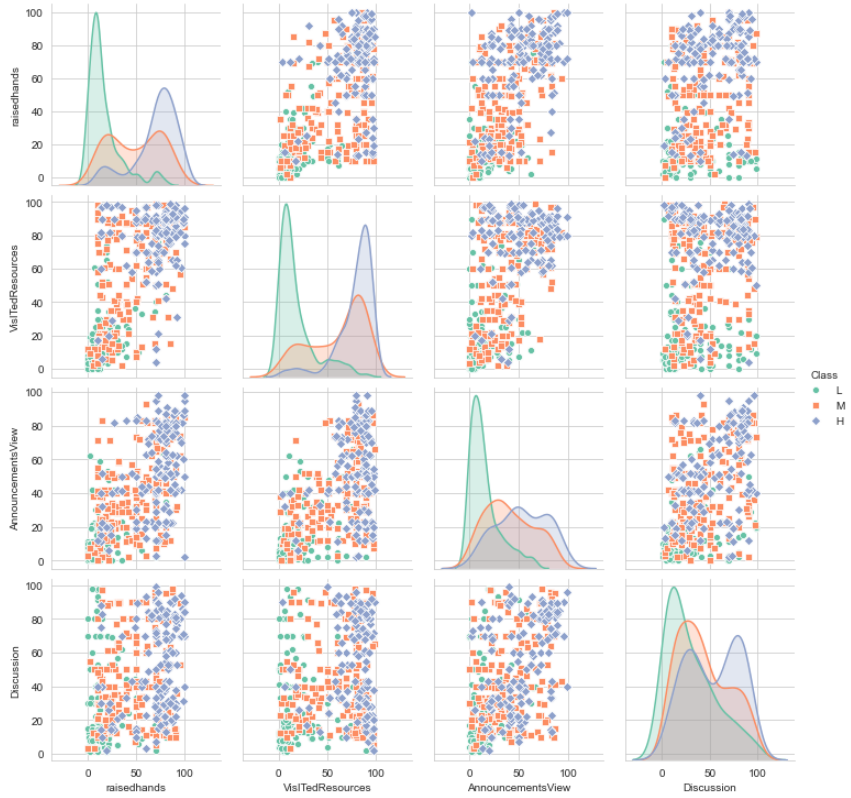
```
1.  fig, axarr  = plt.subplots(2,figsize=(10,10))
2.  axarr[0].set_title('Relation')
3.  axarr[1].set_title('Relation & Academic Performance')
4.  fig.suptitle("The relationship between Students' Academic Performance and Relation"
    , size=20)
5.  sns.countplot(x='Relation', data=data, ax = axarr[0], palette="Paired")
6.  sns.countplot(x='Relation', hue='Class', data=data,hue_order = ['L', 'M', 'H'], ax
    = axarr[1], palette="Set2")
7.  plt.show()
```



根據上兩圖分析，母親作為監護人和學生及格之間似乎有關聯，父親作為監護人和學生不及格之間似乎有關聯。

3.10 學生在課堂舉手次數、訪問課程內容次數、檢查新公告次數、參加討論次數與成績的關係

```
1. sns.pairplot(data, hue="Class",
2.               diag_kind="kde",
3.               hue_order = ['L', 'M', 'H'],
4.               markers=["o", "s", "D"], palette="Set2")
5. plt.show()
```



查看不同教育層次的 raisedhands、VisiTedResources、AnnouncementsView、Discussion，獲得此處中位數：

```
1. data.groupby('GradeID').median()
```

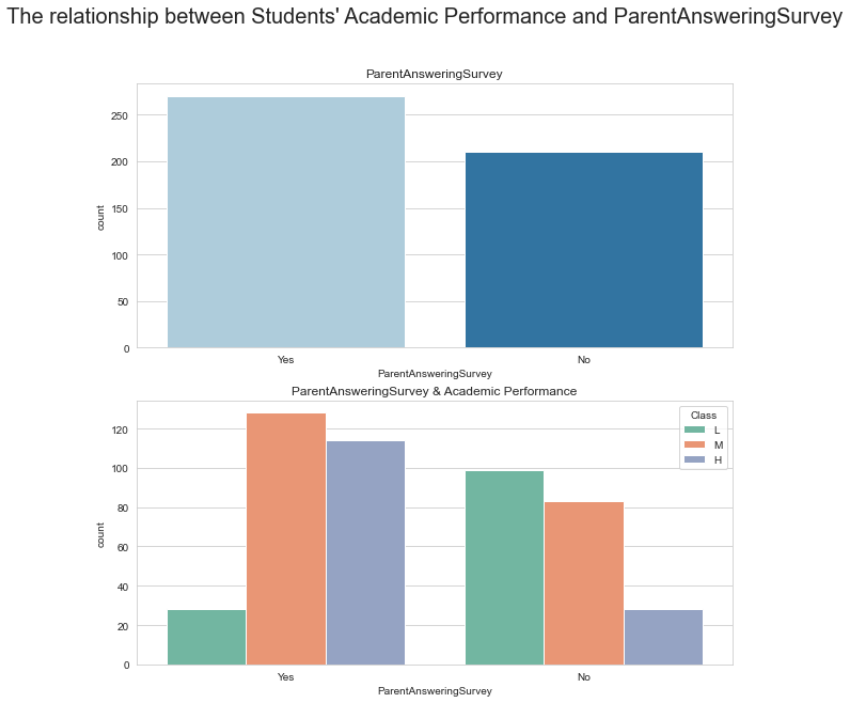
GradeID	raisedhands	VisITedResources	AnnouncementsView	Discussion
G-02	27.0	60.0	21.0	30.0
G-04	45.5	50.0	33.0	43.5
G-05	8.0	10.0	9.0	30.0
G-06	72.0	61.0	49.0	36.5
G-07	50.0	71.0	33.0	50.0
G-08	70.5	77.0	45.5	40.5
G-09	20.0	35.0	30.0	44.0
G-10	33.5	41.5	24.0	26.0
G-11	70.0	63.0	50.0	49.0
G-12	29.0	39.0	19.0	50.0

在這裡我們可以看出五年級和九年級的數據比其他大多數年級少上許多。

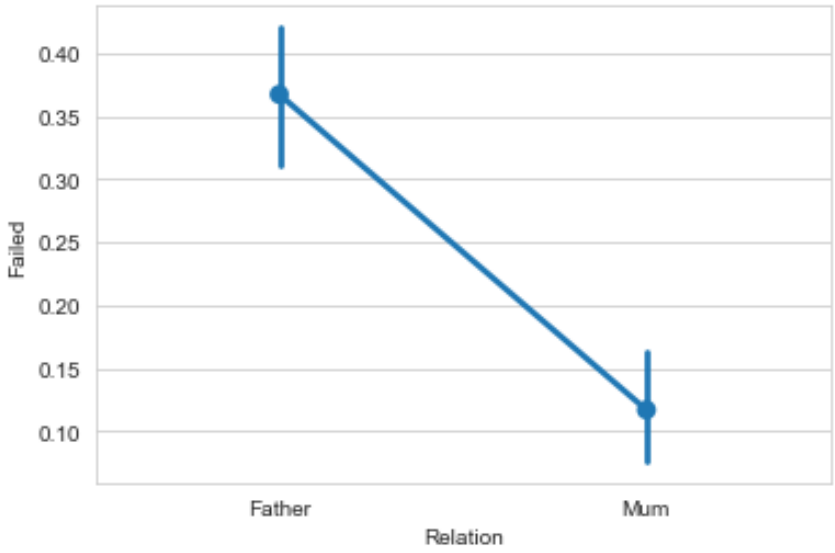
3.11 家長是否回答學校提供的調查與學生成績的關係

```
1. fig, axarr  = plt.subplots(2,figsize=(10,10))
2. axarr[0].set_title('ParentAnsweringSurvey')
3. axarr[1].set_title('ParentAnsweringSurvey & Academic Performance')
```

```
4. fig.suptitle("The relationship between Students' Academic Performance and ParentAnsw
   eringSurvey", size=20)
5. sns.countplot(x='ParentAnsweringSurvey', data=data,
6.                order=['Yes', 'No'], ax = axarr[0], palette="Paired")
7. sns.countplot(x='ParentAnsweringSurvey', hue='Class',
8.                data=data, order=['Yes', 'No'], hue_order = ['L', 'M', 'H'],
9.                ax = axarr[1], palette="Set2")
10. plt.show()
```



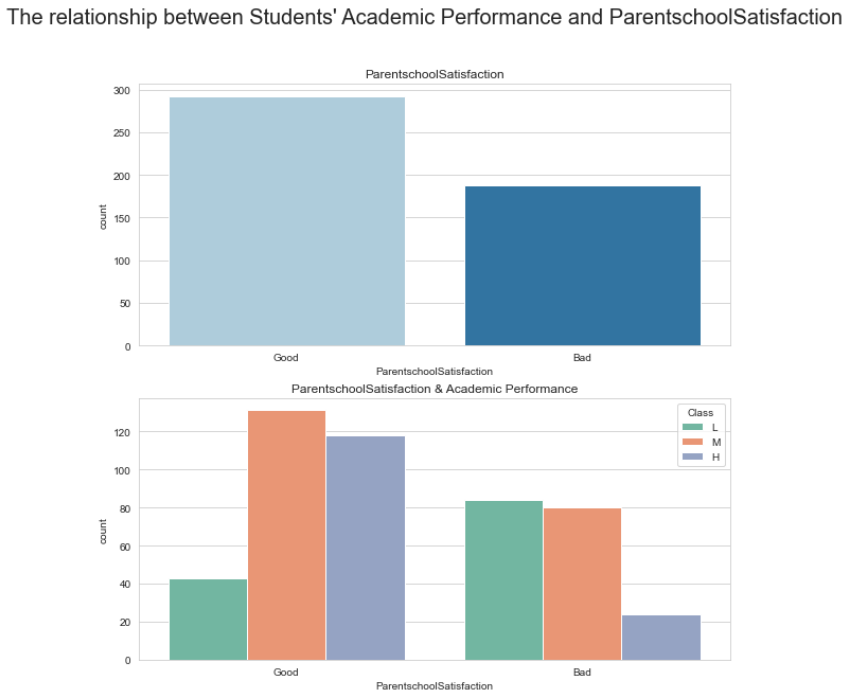

```
1. sns.pointplot(x='Relation', y='Failed', data=data)
```



在成績好的學生中，絕大多數家長對他們所受的教育感到滿意。父母對學校最不滿意的學生的表現要差得多。母親對其負責的學生表現較好。

3.12 家長對於學校滿意度與學生成績的關係

```
1. fig, axarr  = plt.subplots(2,figsize=(10,10))
2. axarr[0].set_title('ParentschoolSatisfaction')
3. axarr[1].set_title('ParentschoolSatisfaction & Academic Performance')
4. fig.suptitle("The relationship between Students' Academic Performance and ParentschoolSatisfaction", size=20)
5. sns.countplot(x='ParentschoolSatisfaction', data=data,
6.                order=['Good', 'Bad'], ax = axarr[0], palette="Paired")
7. sns.countplot(x='ParentschoolSatisfaction', hue='Class',
8.                data=data, order=['Good', 'Bad'],
9.                hue_order = ['L', 'M', 'H'], ax = axarr[1], palette="Set2")
10. plt.show()
```

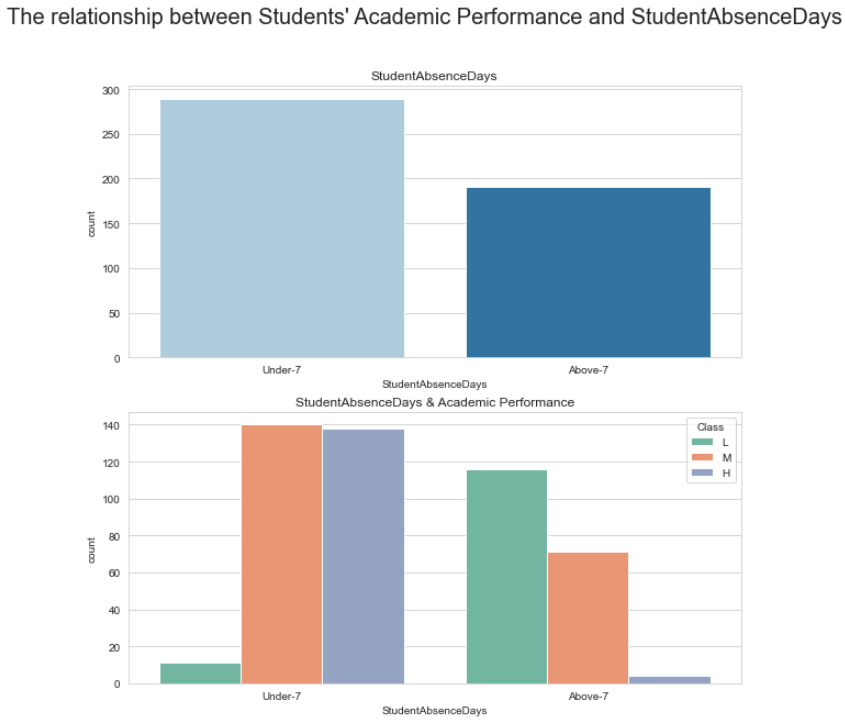


與 3.11 觀察結果相同，不再敘述。

3.13 學生缺勤天數與學生成績的關係

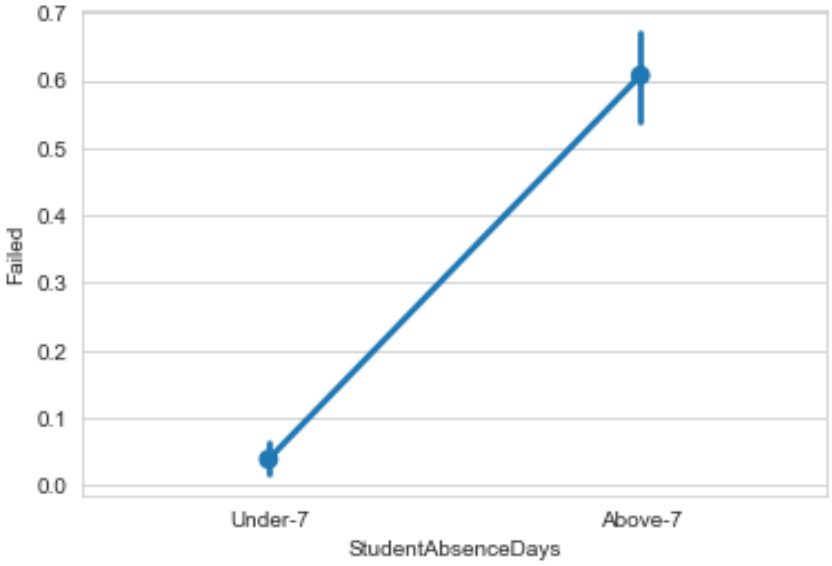
```
1. fig, axarr  = plt.subplots(2,figsize=(10,10))
2. axarr[0].set_title('StudentAbsenceDays')
3. axarr[1].set_title('StudentAbsenceDays & Academic Performance')
4. fig.suptitle("The relationship between Students' Academic Performance and StudentAbsenceDays", size=20)
5. sns.countplot(x='StudentAbsenceDays', data=data,
6.                order=['Under-7', 'Above-7'],
7.                ax = axarr[0], palette="Paired")
8. sns.countplot(x='StudentAbsenceDays', hue='Class',
9.                data=data, order=['Under-7', 'Above-7'],
```

```
10.                 hue_order = ['L', 'M', 'H'],
11.                 ax = axarr[1], palette="Set2")
12. plt.show()
```



根據此圖分析，學習時間與學生成績有很強的相關性，缺課查過七天的學生很少取得高分，缺課少於七天的學生很少不及格。

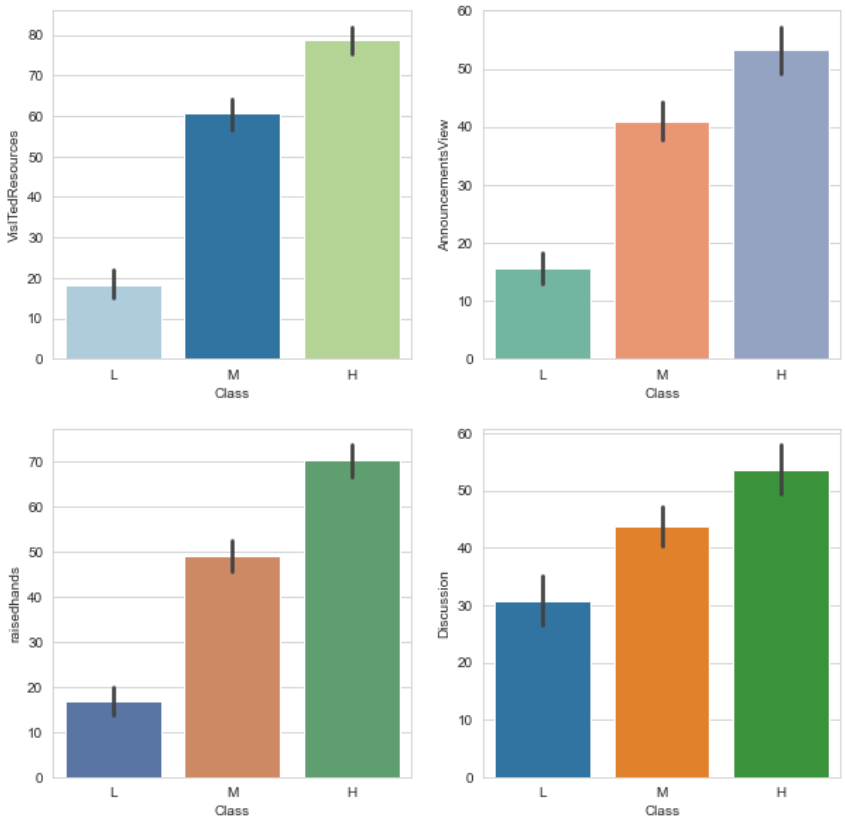
```
1. data['Failed'] = np.where(data['Class']=='L',1,0)
2. sns.pointplot(x='StudentAbsenceDays', y='Failed', data=data)
```



從學生的缺勤頻率可以看出最大的直觀趨勢。基本成績差的學生缺勤次數超過 7 次，而成績好的學生缺勤次數幾乎沒有超過 7 次的。

3.14 學生在課堂舉手次數、訪問課程內容次數、檢查新公告次數、參加討論次數與成績的長條圖

```
1. fig, axarr  = plt.subplots(2,2,figsize=(10,10))
2. sns.barplot(x='Class', y='VisITedResources', data=data, order=['L','M','H'], ax=axar
   r[0,0], palette="Paired")
3. sns.barplot(x='Class', y='AnnouncementsView', data=data, order=['L','M','H'], ax=axa
   rr[0,1], palette="Set2")
4. sns.barplot(x='Class', y='raisedhands', data=data, order=['L','M','H'], ax=axarr[1,0
   ], palette="deep")
5. sns.barplot(x='Class', y='Discussion', data=data, order=['L','M','H'], ax=axarr[1,1]
   , palette="tab10")
```

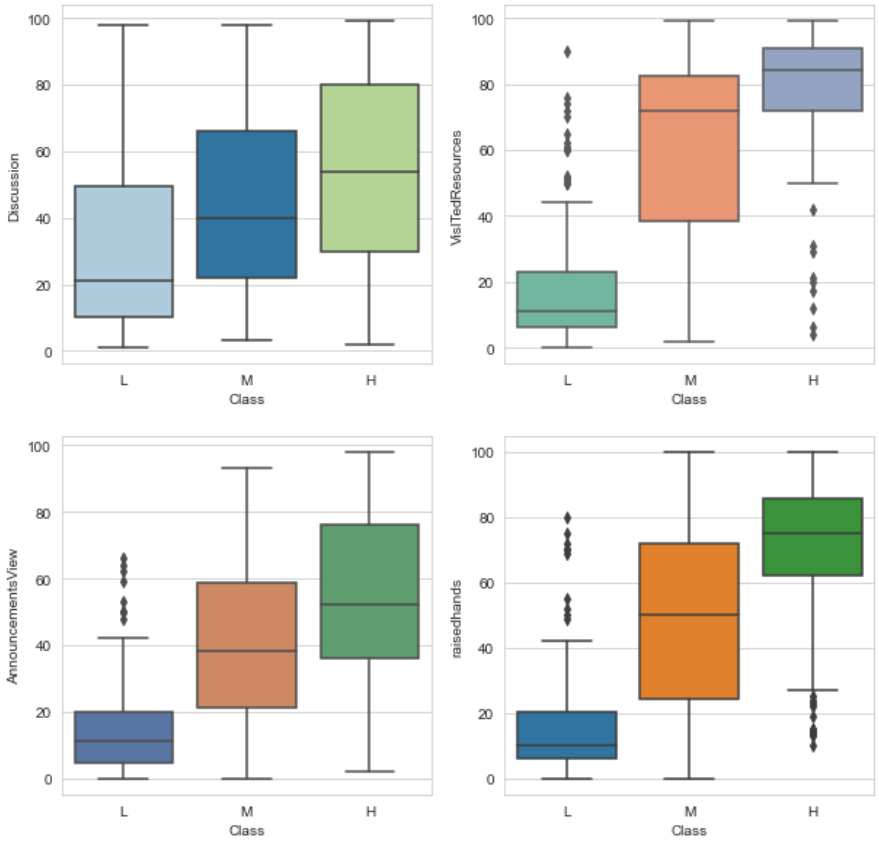


正如預期的那樣，那些參與較多的人（討論、舉手、公告瀏覽、舉手次數較多），表現較好.....這就是相關性和因果性的事情。

3.15 課堂活躍度對比

```
1. fig, axarr = plt.subplots(2, 2,figsize=(10,10))
2. sns.boxplot(x='Class', y='Discussion', data=data, order=['L','M','H'], ax=axarr[0,0]
   , palette="Paired")
3. sns.boxplot(x='Class', y='VisITedResources', data=data, order=['L','M','H'], ax=axar
   r[0,1], palette="Set2")
4. sns.boxplot(x='Class', y='AnnouncementsView', data=data, order=['L','M','H'], ax=axa
   rr[1,0], palette="deep")
```

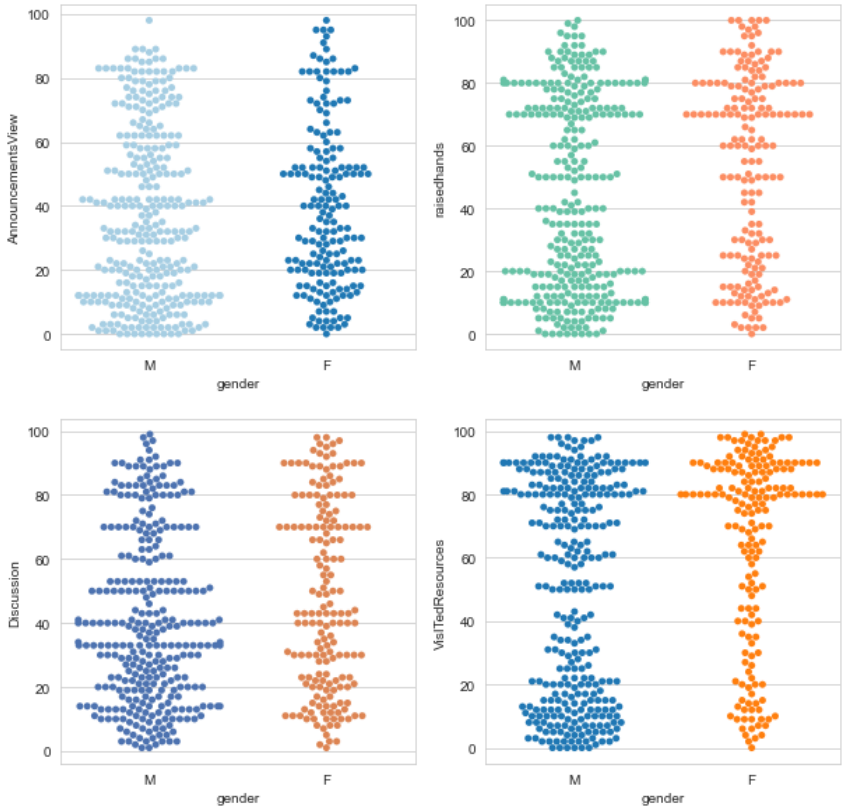
```
5. sns.boxplot(x='Class', y='raisedhands', data=data, order=['L','M','H'], ax=axarr[1,1
   ], palette="tab10")
```



根據上圖分析，訪問課程內容可能並不像討論那樣是表現良好的必經之路，舉手可能並不像檢查新公告那樣是表現良好的必經之路。

3.16 性別和課堂參與的對比

```
1. fig, axarr = plt.subplots(2, 2,figsize=(10,10))
2. sns.swarmplot(x='gender', y='AnnouncementsView', data=data, ax=axarr[0,0], palette="
   Paired")
3. sns.swarmplot(x='gender', y='raisedhands', data=data, ax=axarr[0,1], palette="Set2")
4. sns.swarmplot(x='gender', y='Discussion', data=data, ax=axarr[1,0], palette="deep")
5. sns.swarmplot(x='gender', y='VisITedResources', data=data, ax=axarr[1,1], palette="t
   ab10")
```



這個蜂群圖告訴我們，獲得低分（L）的學生比獲得 M 或 H 分的學生訪問的資源要熱得多。此外，獲得高分（H）的女性幾乎只訪問了大量的在線資源。

綜上分析，學生成績與訪問課程內容的次數、缺席天數、在課上有舉手的次數、檢查新公告的次數、是否參加討論、性別、監護人、學期這些屬性有關。

4. 建立模型預測與學生成績的關係

4.1 處理資料

```
# Convert grades into data
gradeID_dict = {"G-01" : 1,
                "G-02" : 2,
                "G-03" : 3,
                "G-04" : 4,
                "G-05" : 5,
                "G-06" : 6,
                "G-07" : 7,
                "G-08" : 8,
                "G-09" : 9,
                "G-10" : 10,
                "G-11" : 11,
                "G-12" : 12}

data = data.replace({"GradeID" : gradeID_dict})
# Convert scores into data
```

```
class_dict = {"L" : -1,
                "M" : 0,
                "H" : 1}
data = data.replace({"Class" : class_dict})

# Convert to Scale data
data["GradeID"] = preprocessing.scale(data["GradeID"])
data["raisedhands"] = preprocessing.scale(data["raisedhands"])
data["VisITedResources"] = preprocessing.scale(data["VisITedResources"])
data["AnnouncementsView"] = preprocessing.scale(data["AnnouncementsView"])
data["Discussion"] = preprocessing.scale(data["Discussion"])

# Use virtual code conversion to convert 11 columns into 64 columns
data = pd.get_dummies(data, columns=["gender",
                                     "NationalITy",
                                     "PlaceofBirth",
                                     "SectionID",
                                     "StageID",
                                     "Topic",
                                     "Semester",
                                     "Relation",
                                     "ParentAnsweringSurvey",
                                     "ParentschoolSatisfaction",
                                     "StudentAbsenceDays"])


data.head()
```

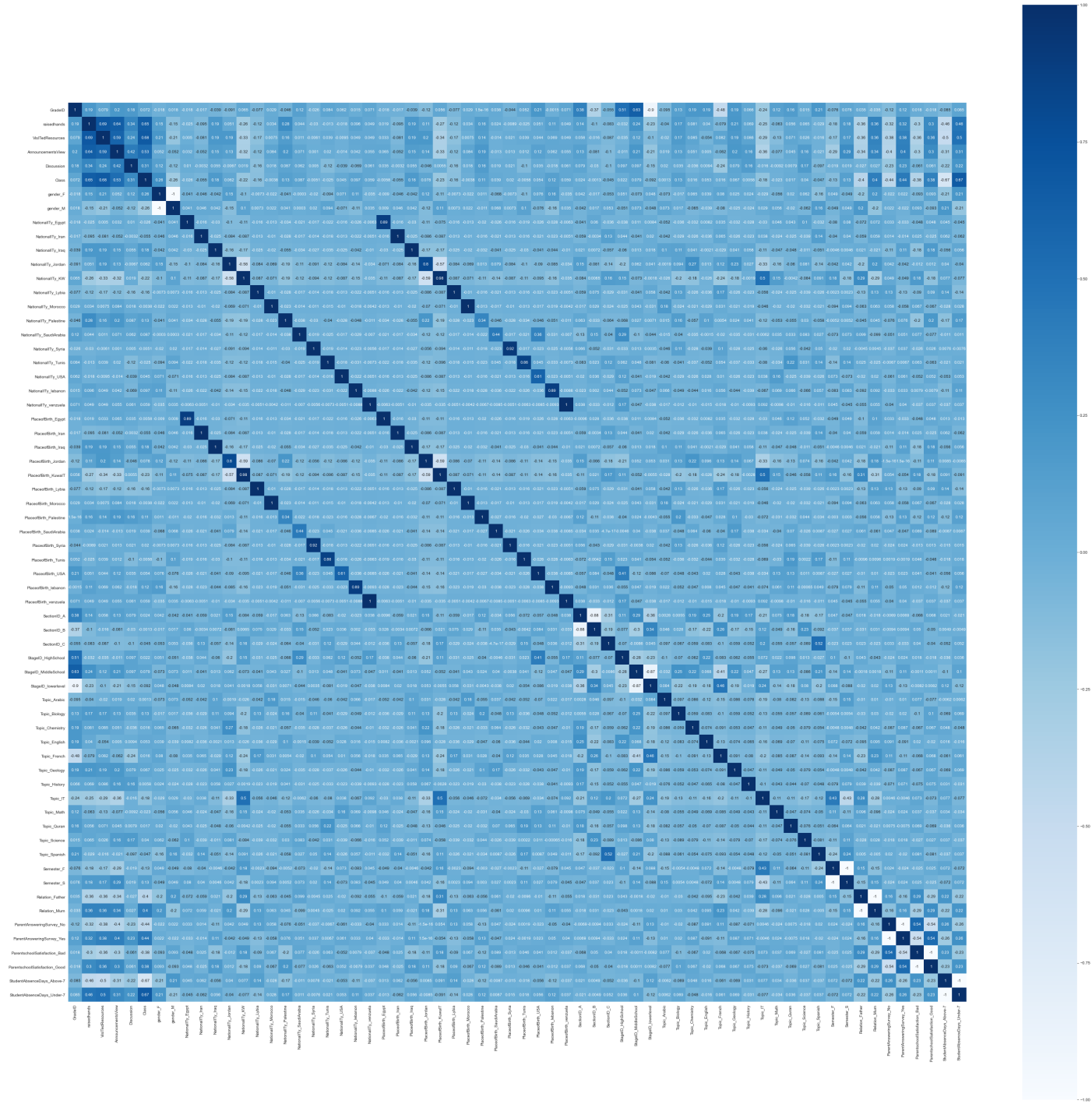
	GradeID	raisedhands	VisITedResources	AnnouncementsView	Discussion	Class	gender_F	gender_M	NationalITy_Egypt	NationalITy_Iran	...	Semester_F
0	-0.563838	-1.033429	-1.174075	-1.351167	-0.843326	0	0	1	0	0	...	1
1	-0.563838	-0.870813	-1.053029	-1.313549	-0.662225	0	0	1	0	0	...	1
2	-0.563838	-1.196046	-1.446426	-1.426401	-0.481125	-1	0	1	0	0	...	1
3	-0.563838	-0.545579	-0.901723	-1.238315	-0.300024	-1	0	1	0	0	...	1
4	-0.563838	-0.220346	-0.145191	-0.974994	0.243279	0	0	1	0	0	...	1

5 rows × 64 columns

4.2 列出成績與其他屬性的相關性

```
corr = data.corr()
mask = np.triu(np.ones_like(corr, dtype=bool))
f, ax = plt.subplots(figsize=(11, 9))
cmap = sns.diverging_palette(230, 20, as_cmap=True)
sns.heatmap(corr, mask=mask, cmap=cmap, vmax=.3, center=0,
```

```
6.                square=True, linewidths=.5, cbar_kws={"shrink": .5})
```



我們單獨查看 Class 與其他列的關係：

```
1. corr = data.corr()
2. corr.iloc[[5]]
```

Out[51]:

	GradeID	raisedhands	VisITedResources	AnnouncementsView	Discussion	Class	gender_F	gender_M	NationalITy_Egypt	NationalITy_Iran	...	Semeste
Class	0.071654	0.646298	0.677094	0.52737	0.308183	1.0	0.26349	-0.26349	-0.02631	-0.054841	...	-0.1262

1 rows × 64 columns

根據上圖和表格，我們可以看出訪問課程內容的次數、缺席天數、在課上有舉手的次數、檢查新公告的次數、是否參加討論、性別、監護人、學期都與 Class 有很強的相關性，這和我們之前的分析一樣。

5. 訓練與預測

5.1 找出預測準確度最高的分類器

```
X = data.drop('Class', axis=1)
y = data['Class']

# Encoding our categorical columns in X
labelEncoder = LabelEncoder()
cat_columns = X.dtypes.pipe(lambda x: x[x == 'object']).index
for col in cat_columns:
    X[col] = labelEncoder.fit_transform(X[col])

# Train Test Split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=52)

# Create the radial basis function kernel version of a Support Vector Machine classifier
rbf_clf = svm.SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
             decision_function_shape='ovo', degree=3, gamma='auto', kernel='rbf',
             max_iter=-1, probability=False, random_state=None, shrinking=True,
             tol=0.001, verbose=False)
# Create the linear kernel version of a Support Vector Machine classifier
lin_clf = svm.SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
             decision_function_shape='ovo', degree=3, gamma='auto', kernel='linear',
             max_iter=-1, probability=False, random_state=None, shrinking=True,
             tol=0.001, verbose=False)
# Create the polynomial kernel version of a Support Vector Machine classifier
poly_clf = svm.SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
             decision_function_shape='ovo', degree=3, gamma='auto', kernel='poly',
             max_iter=-1, probability=False, random_state=None, shrinking=True,
             tol=0.001, verbose=False)
# Create the sigmoid kernel version of a Support Vector Machine classifier
sig_clf = svm.SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
             decision_function_shape='ovo', degree=3, gamma='auto', kernel='sigmoid',
```

```
                max_iter=-1, probability=False, random_state=None, shrinking=True,
                tol=0.001, verbose=False)
keys = []
scores = []
models = {'Logistic Regression': LogisticRegression(max_iter=3000), 'Decision Tree':
    DecisionTreeClassifier(),
           'Random Forest': RandomForestClassifier(n_estimators=300, random_state=52)
   ,'Perceptron':Perceptron(eta0=0.1, random_state=15),'RBF':rbf_clf,'Linear':lin_clf,

          'Polynomial':poly_clf,'Sigmoid':sig_clf}

for k,v in models.items():
    mod = v
    mod.fit(X_train, y_train)
    pred = mod.predict(X_test)
    print('Results for: ' + str(k) + '\n')
    print(confusion_matrix(y_test, pred))
    print(classification_report(y_test, pred))
    acc = accuracy_score(y_test, pred)
    print("accuracy is "+ str(acc))
    print('\n' + '\n')
    keys.append(k)
    scores.append(acc)
    table = pd.DataFrame({'model':keys, 'accuracy score':scores})

print(table)
```

Output:

```
Results for: Logistic Regression

[[28  7  1]
 [ 4 43  9]
 [ 0 12 40]]
              precision    recall  f1-score   support

          -1       0.88      0.78      0.82        36
           0       0.69      0.77      0.73        56
           1       0.80      0.77      0.78        52

    accuracy                           0.77       144
   macro avg       0.79      0.77      0.78       144
weighted avg       0.78      0.77      0.77       144
```

```
15.
16.accuracy is 0.7708333333333334
17.
18.
19.
20.Results for: Decision Tree
21.
22.[[29  6  1]
23. [ 4 39 13]
24. [ 0 21 31]]
25.              precision    recall  f1-score   support
26.
27.          -1       0.88      0.81      0.84        36
28.           0       0.59      0.70      0.64        56
29.           1       0.69      0.60      0.64        52
30.
31.    accuracy                           0.69       144
32.   macro avg       0.72      0.70      0.71       144
33.weighted avg       0.70      0.69      0.69       144
34.
35.accuracy is 0.6875
36.
37.
38.
39.Results for: Random Forest
40.
41.[[31  4  1]
42. [ 3 49  4]
43. [ 0 12 40]]
44.              precision    recall  f1-score   support
45.
46.          -1       0.91      0.86      0.89        36
47.           0       0.75      0.88      0.81        56
48.           1       0.89      0.77      0.82        52
49.
50.    accuracy                           0.83       144
51.   macro avg       0.85      0.84      0.84       144
52.weighted avg       0.84      0.83      0.83       144
53.
54.accuracy is 0.8333333333333334
55.
56.
```

```
Results for: Perceptron

[[33  2  1]
 [16 34  6]
 [ 3 25 24]]
              precision    recall  f1-score   support

          -1       0.63      0.92      0.75        36
           0       0.56      0.61      0.58        56
           1       0.77      0.46      0.58        52

    accuracy                           0.63       144
   macro avg       0.66      0.66      0.64       144
weighted avg       0.65      0.63      0.62       144

accuracy is 0.6319444444444444


Results for: RBF

[[32  4  0]
 [ 5 48  3]
 [ 0 14 38]]
              precision    recall  f1-score   support

          -1       0.86      0.89      0.88        36
           0       0.73      0.86      0.79        56
           1       0.93      0.73      0.82        52

    accuracy                           0.82       144
   macro avg       0.84      0.83      0.83       144
weighted avg       0.83      0.82      0.82       144

accuracy is 0.8194444444444444


Results for: Linear

[[29  6  1]
```

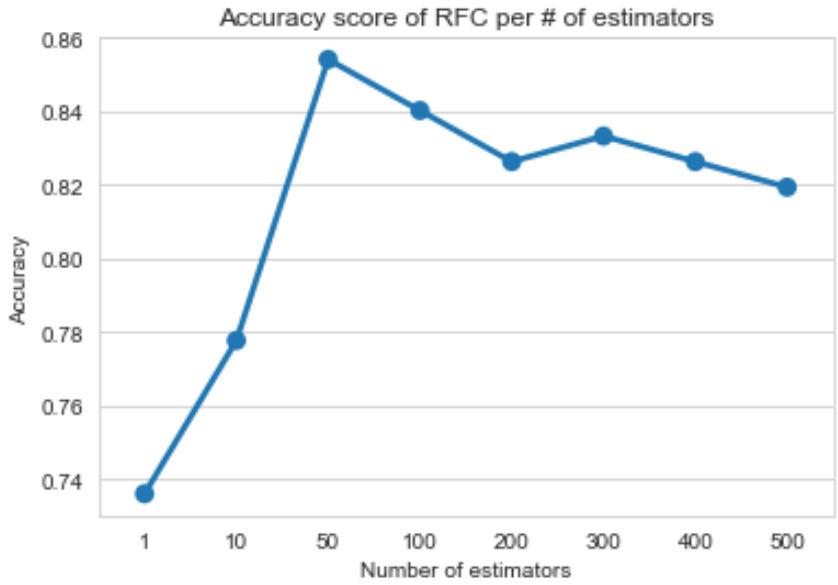
```
[ 4 43  9]
 [ 0 12 40]]
              precision    recall  f1-score   support

          -1       0.88      0.81      0.84        36
           0       0.70      0.77      0.74        56
           1       0.80      0.77      0.78        52

    accuracy                           0.78       144
   macro avg       0.79      0.78      0.79       144
weighted avg       0.78      0.78      0.78       144

accuracy is 0.7777777777777778


Results for: Polynomial

[[ 0 36  0]
 [ 0 56  0]
 [ 0 52  0]]
              precision    recall  f1-score   support

          -1       0.00      0.00      0.00        36
           0       0.39      1.00      0.56        56
           1       0.00      0.00      0.00        52

    accuracy                           0.39       144
   macro avg       0.13      0.33      0.19       144
weighted avg       0.15      0.39      0.22       144

accuracy is 0.3888888888888889


Results for: Sigmoid

[[32  4  0]
 [ 6 46  4]
 [ 1 18 33]]
              precision    recall  f1-score   support
```

```
           -1       0.82      0.89      0.85        36
            0       0.68      0.82      0.74        56
            1       0.89      0.63      0.74        52

     accuracy                           0.77       144
    macro avg       0.80      0.78      0.78       144
weighted avg       0.79      0.77      0.77       144

accuracy is 0.7708333333333334


                  model  accuracy score
0  Logistic Regression        0.770833
1        Decision Tree        0.687500
2        Random Forest        0.833333
3           Perceptron        0.631944
4                  RBF        0.819444
5               Linear        0.777778
6           Polynomial        0.388889
7              Sigmoid        0.770833
```

如上表可見，Random Forest 是預測最準確的分類器，準確率高達 83.3%。讓我們進一步探討森林中 estimators 的數量。一個普遍的規則是，當 estimators 數量增加時，這個分類器表現更好。

5.2 探索調優 Random Forest 分類器

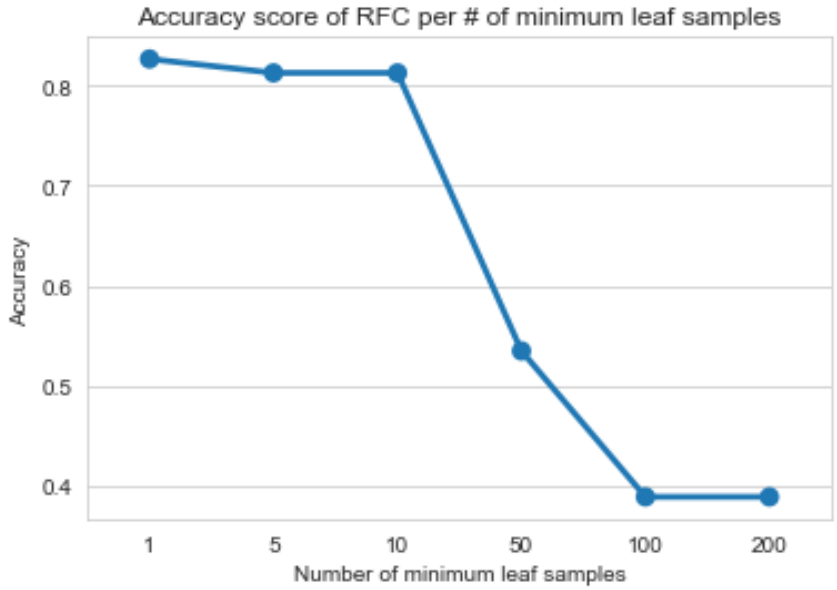
```
# Exploring the number of estimators in the random forest
score = []
est = []
estimators = [1, 10, 50, 100, 200, 300, 400, 500]
for e in estimators:
    rfc1 = RandomForestClassifier(n_estimators=e, random_state=52)
    pred1 = rfc1.fit(X_train, y_train).predict(X_test)
    accuracy = accuracy_score(y_test, pred1)
    score.append(accuracy)
    est.append(e)
plot = sns.pointplot(x=est, y=score)
plot.set(xlabel='Number of estimators', ylabel='Accuracy',
         title='Accuracy score of RFC per # of estimators')
plt.show()
```



事實上，當 estimators 的數量增加時，RFC 的表現更好。然而，在 200 個 estimators 時，它就會趨於平穩。顯然，200 個 estimators 對於這個資料集來說已經足夠了。

我們還可以探索另一個變數，比如一個葉子節點所需的最小樣本數。

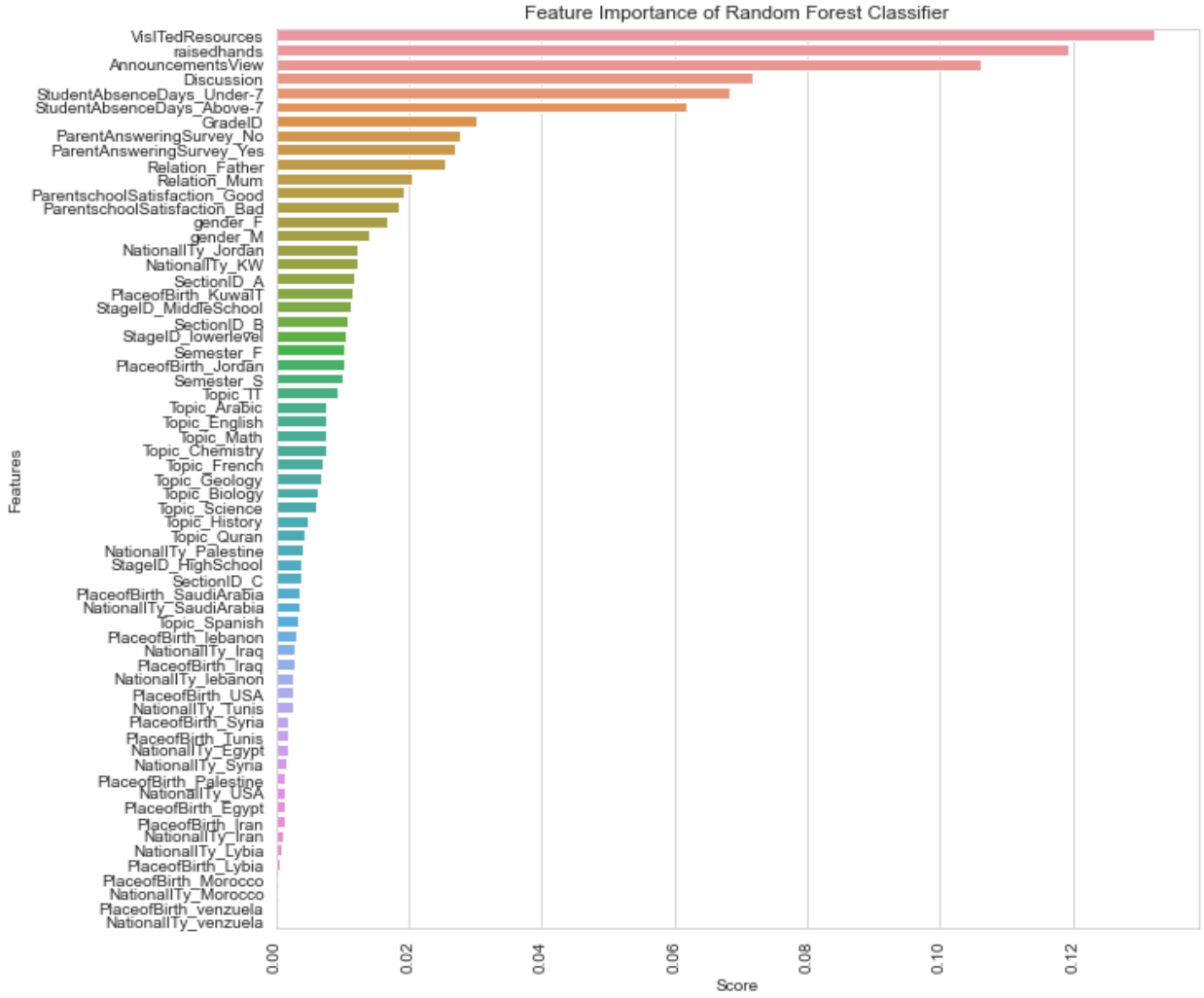
```
1. # Exploring minimum leaf samples
2. score = []
3. leaf = []
4. leaf_options = [1, 5, 10, 50, 100, 200]
5. for l in leaf_options:
6.     rfc2 = RandomForestClassifier(n_estimators=200, random_state=52, min_samples_lea
   f=l)
7.     pred2 = rfc2.fit(X_train, y_train).predict(X_test)
8.     accuracy = accuracy_score(y_test, pred2)
9.     score.append(accuracy)
10.    leaf.append(l)
11.plot = sns.pointplot(x=leaf, y=score)
12.plot.set(xlabel='Number of minimum leaf samples', ylabel='Accuracy',
13.         title='Accuracy score of RFC per # of minimum leaf samples')
14.plt.show()
```



在這種情況下，我們可以看到，隨著最小葉子樣本的增加，準確率分數會簡單地降低。因此，最好將該值保持在預設的 1。

讓我們來評估 RFC 的特徵重要性。

```
1. rfc = RandomForestClassifier(n_estimators=200, random_state=52)
2. pred = rfc.fit(X_train, y_train).predict(X_test)
3. dn = {'features':X.columns, 'score':rfc.feature_importances_}
4. df = pd.DataFrame.from_dict(data=dn).sort_values(by='score', ascending=False)
5. plot = sns.barplot(x='score', y='features', data=df, orient='h')
6. plot.set(xlabel='Score', ylabel='Features',
7.          title='Feature Importance of Random Forest Classifier')
8. plt.rcParams['figure.figsize']=(20,20)
9. plt.setp(plot.get_xticklabels(), rotation=90)
10.plt.show()
```



訪問課程內容的次數是最重要的特徵。

結論：

從我們的得出的結果來看，訪問課程內容的次數、缺席天數、在課上有舉手的次數、檢查新公告的次數、是否參加討論、性別、監護人、學期確確實實是影響學生學業成績的因素。