In [1]:
```
import numpy as np
import pandas as pd
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
```

In [2]:
```
dataset = pd.read_csv('college.csv')
dataset.head()
```

Out[2]:

	StudentID	Gender	Parent_income	IQ	Encourage	Plan
0	4558	male	53900	118	encourage	plan
1	4561	female	24900	87	not encourage	not plan
2	4563	female	65800	93	not encourage	not plan
3	4565	male	11440	117	encourage	plan
4	4567	female	16700	102	not encourage	not plan

In [16]:
```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
dataset['Gender'] = le.fit_transform(dataset['Gender'])
dataset.head(100)
```

Out[16]:

	StudentID	Gender	Parent_income	IQ	Encourage	Plan
0	4558	1	53900	118	encourage	1
1	4561	0	24900	87	not encourage	0
2	4563	0	65800	93	not encourage	0
3	4565	1	11440	117	encourage	1
4	4567	0	16700	102	not encourage	0
...	...	...	...	...	...	...
95	4742	0	49280	104	encourage	1
96	4744	0	32670	129	encourage	0
97	4748	1	50380	107	encourage	1
98	4750	0	11000	132	not encourage	0
99	4752	0	39600	74	encourage	0

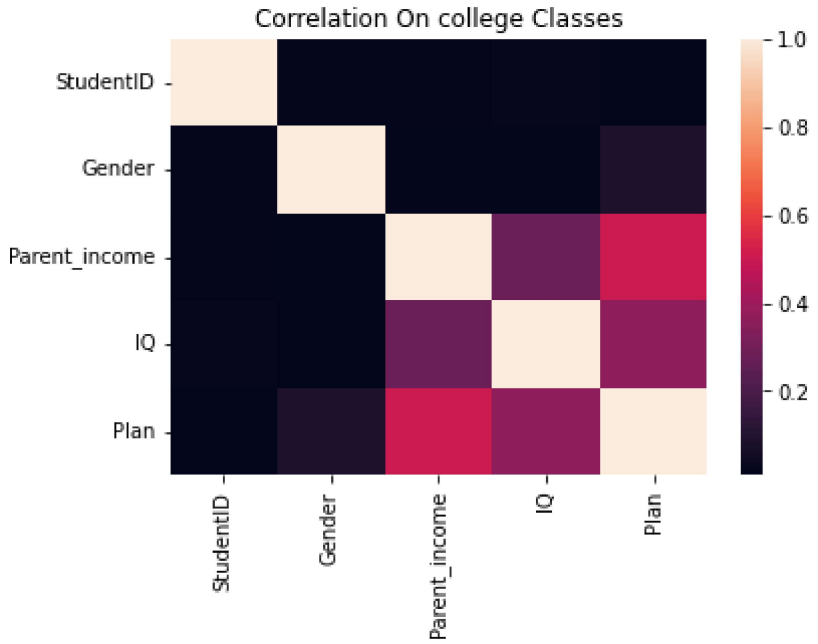
100 rows × 6 columns

In [17]:
```
import matplotlib.pyplot as plt
```

In [18]:
```
import seaborn as sns
```

In [19]:
```
plt.figure(1)
sns.heatmap(dataset.corr())
plt.title('Correlation On college Classes')
```

Out[19]: Text(0.5, 1.0, 'Correlation On college Classes')



In [20]:
```
from sklearn.model_selection import train_test_split
```

In [21]:
```
X = dataset.drop(columns = ['Plan'])
Y = dataset['Plan']
X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 0.25)
```

In [22]:
```
dummy = pd.get_dummies(dataset['Gender'])
```

In [23]:
```
dummy.head()
```

Out[23]:

	0	1
0	0	1
1	1	0
2	1	0
3	0	1
4	1	0

In [24]:
```

X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 0.25)
```

In [26]:
```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
dataset['Encourage'] = le.fit_transform(dataset['Encourage'])
dataset.head(100)
```

Out[26]:

	StudentID	Gender	Parent_income	IQ	Encourage	Plan
0	4558	1	53900	118	0	1
1	4561	0	24900	87	1	0
2	4563	0	65800	93	1	0
3	4565	1	11440	117	0	1
4	4567	0	16700	102	1	0
...	...	...	...	...	...	...
95	4742	0	49280	104	0	1
96	4744	0	32670	129	0	0
97	4748	1	50380	107	0	1
98	4750	0	11000	132	1	0
99	4752	0	39600	74	0	0

100 rows × 6 columns

In [28]:
```
X = dataset.drop(columns = ['Plan'])
Y = dataset['Plan']
X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 0.25)
```

In [29]:
```
classifier = DecisionTreeClassifier()
classifier.fit(X_train, Y_train)
y_pred = classifier.predict(X_test)

print(classification_report(Y_test, y_pred))
print(confusion_matrix(Y_test, y_pred))

print('accuracy is'), accuracy_score(y_pred,Y_test)
```

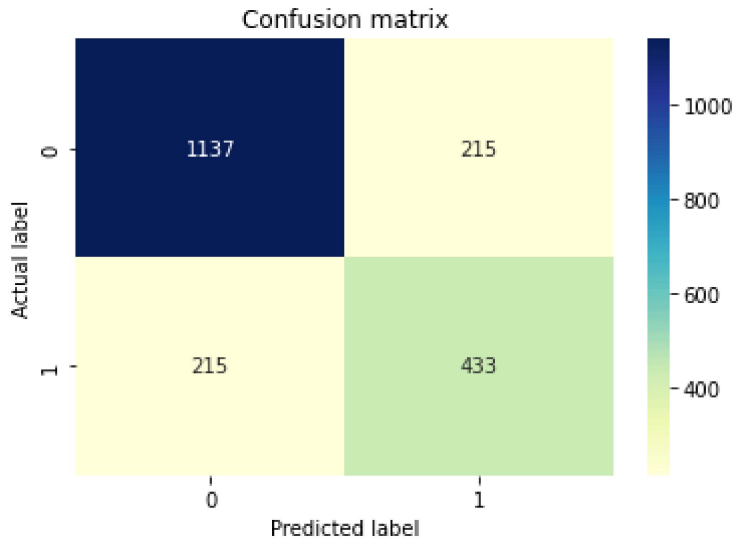
```
              precision    recall  f1-score   support

           0       0.84      0.84      0.84      1352
           1       0.67      0.67      0.67       648

    accuracy                           0.79      2000
   macro avg       0.75      0.75      0.75      2000
weighted avg       0.79      0.79      0.78      2000

[[1137  215]
 [ 215  433]]
accuracy is
```

Out[29]: (None, 0.785)

In [32]:
```
from sklearn import metrics
cnf_matrix = metrics.confusion_matrix(Y_test, y_pred)
p = sns.heatmap(pd.DataFrame(cnf_matrix), annot=True, cmap="YlGnBu" ,fmt='g')
plt.title('Confusion matrix', Y=1.1)
plt.ylabel('Actual label')
plt.xlabel('Predicted label')
```

```
C:\Users\21f22141\AppData\Local\Temp/ipykernel_19164/3092745772.py:4: Matplotli
bDeprecationWarning: Case-insensitive properties were deprecated in 3.3 and sup
port will be removed two minor releases later
  plt.title('Confusion matrix', Y=1.1)
```

Out[32]: Text(0.5, 15.0, 'Predicted label')



In [ ]: