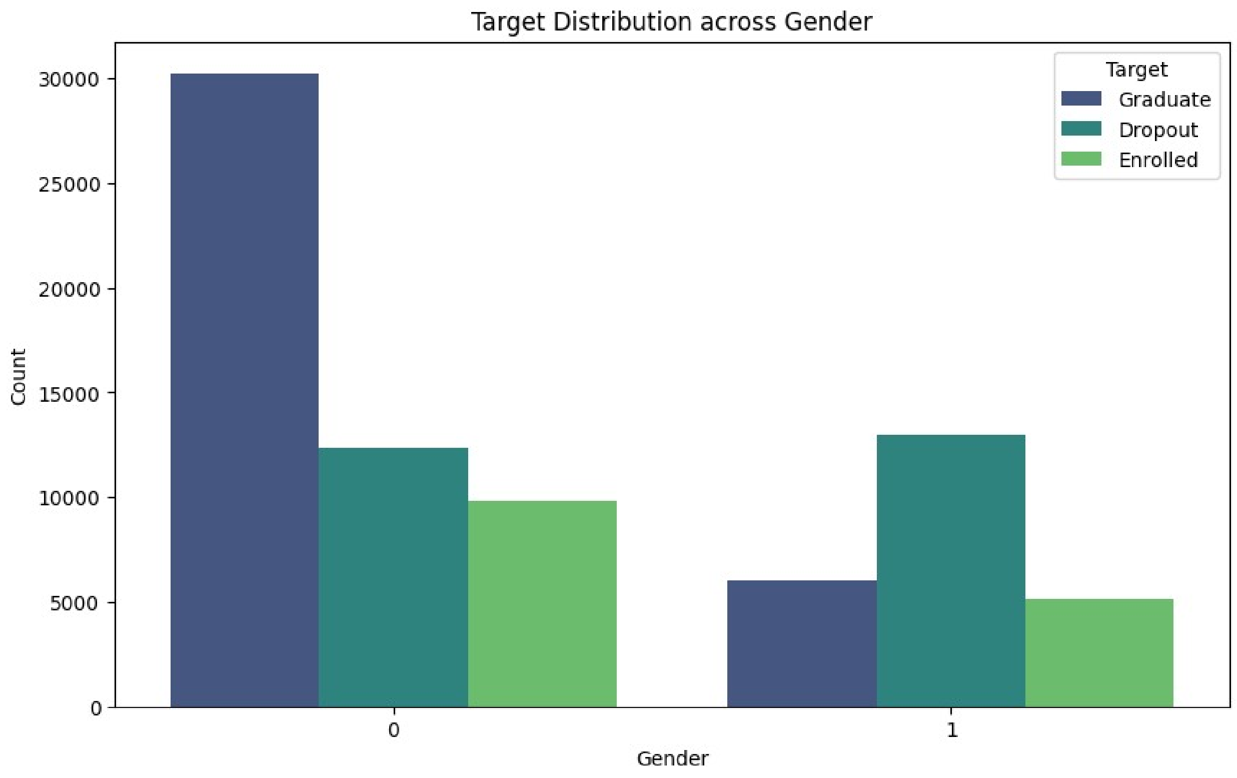
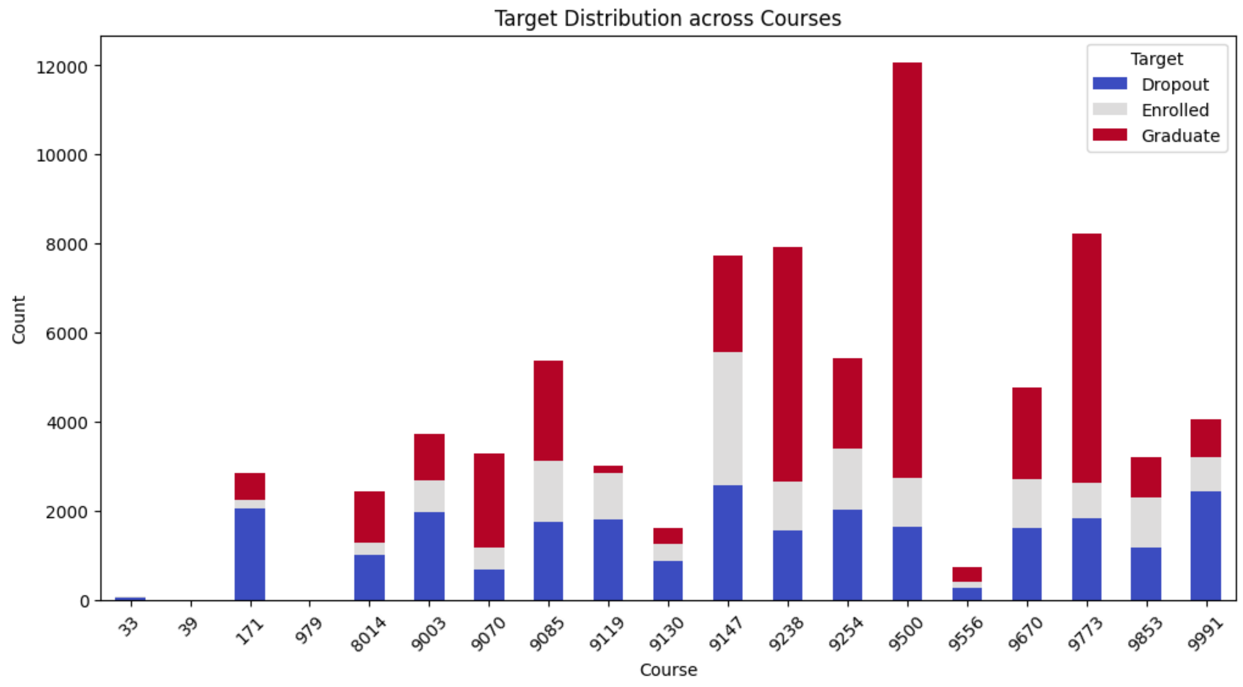
Academic Success Classification Model

Exploratory Data Analysis

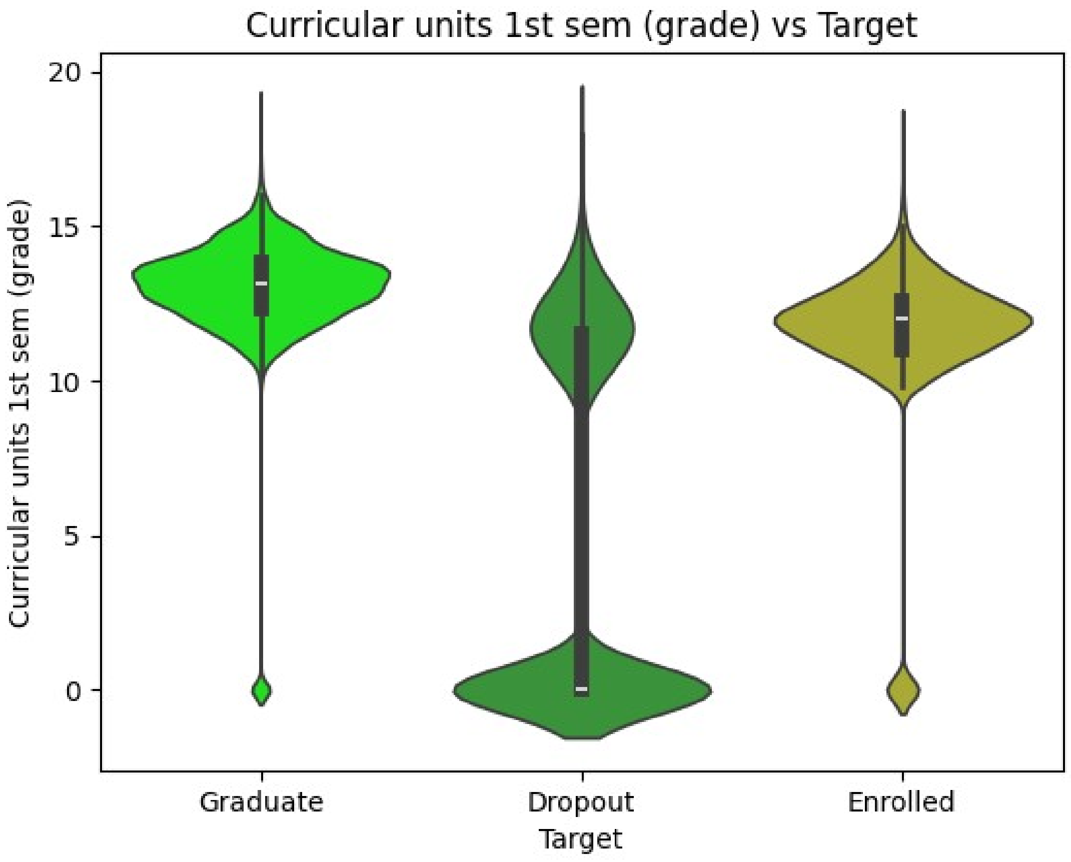
```python
# Importing Needed Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

train_data = pd.read_csv('train.csv')
test_data = pd.read_csv('test.csv')

# checking the target distribution across gender
plt.figure(figsize=(10,6))
sns.countplot(data=train_data, x='Gender', hue='Target',
palette='viridis')
plt.title("Target Distribution across Gender")
plt.xlabel('Gender')
plt.ylabel('Count')
plt.legend(title='Target', loc='upper right')
plt.show()
```

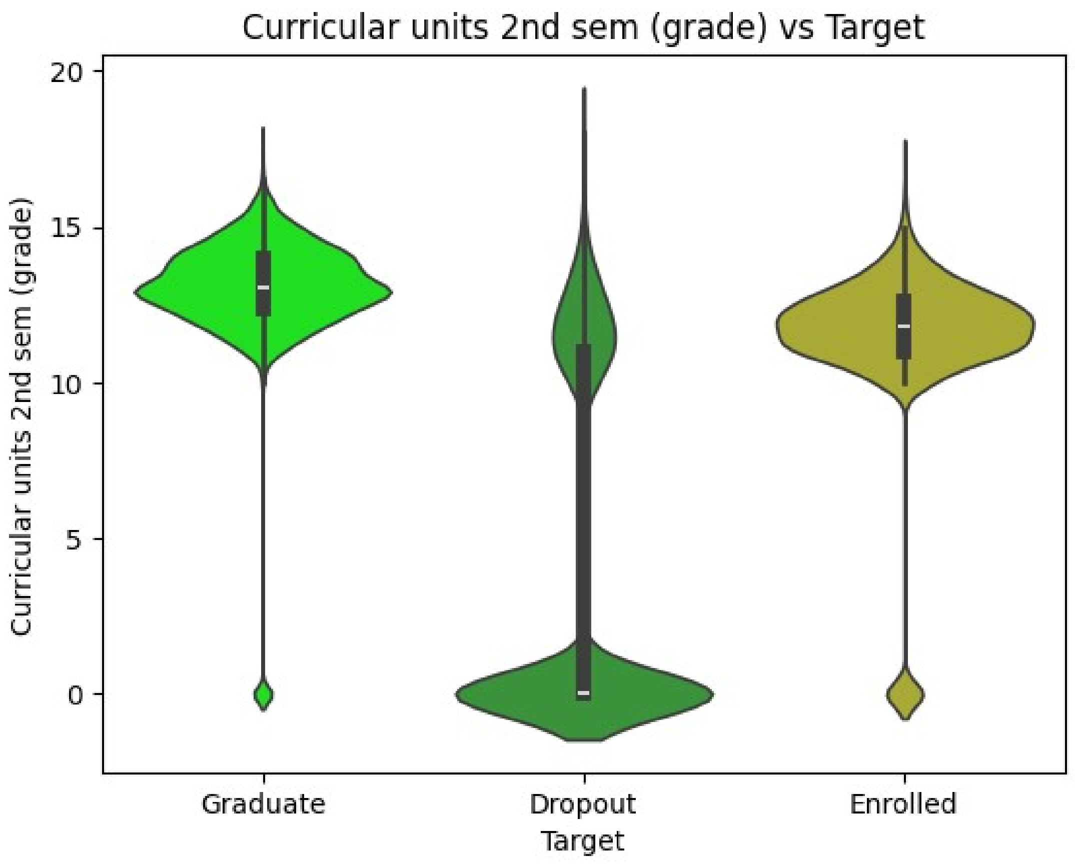


```python
# Calculate the count of Target categories for each course
coursewise_Target = train_data.groupby(['Course',
'Target']).size().unstack(fill_value=0)

# plot Stacked bar chart
coursewise_Target.plot(kind='bar', stacked=True, figsize=(12,6),
colormap= 'coolwarm')
plt.title('Target Distribution across Courses')
plt.xlabel('Course')
plt.ylabel('Count')
plt.legend(title='Target', loc='upper right')
plt.xticks(rotation=45)
plt.show()
```

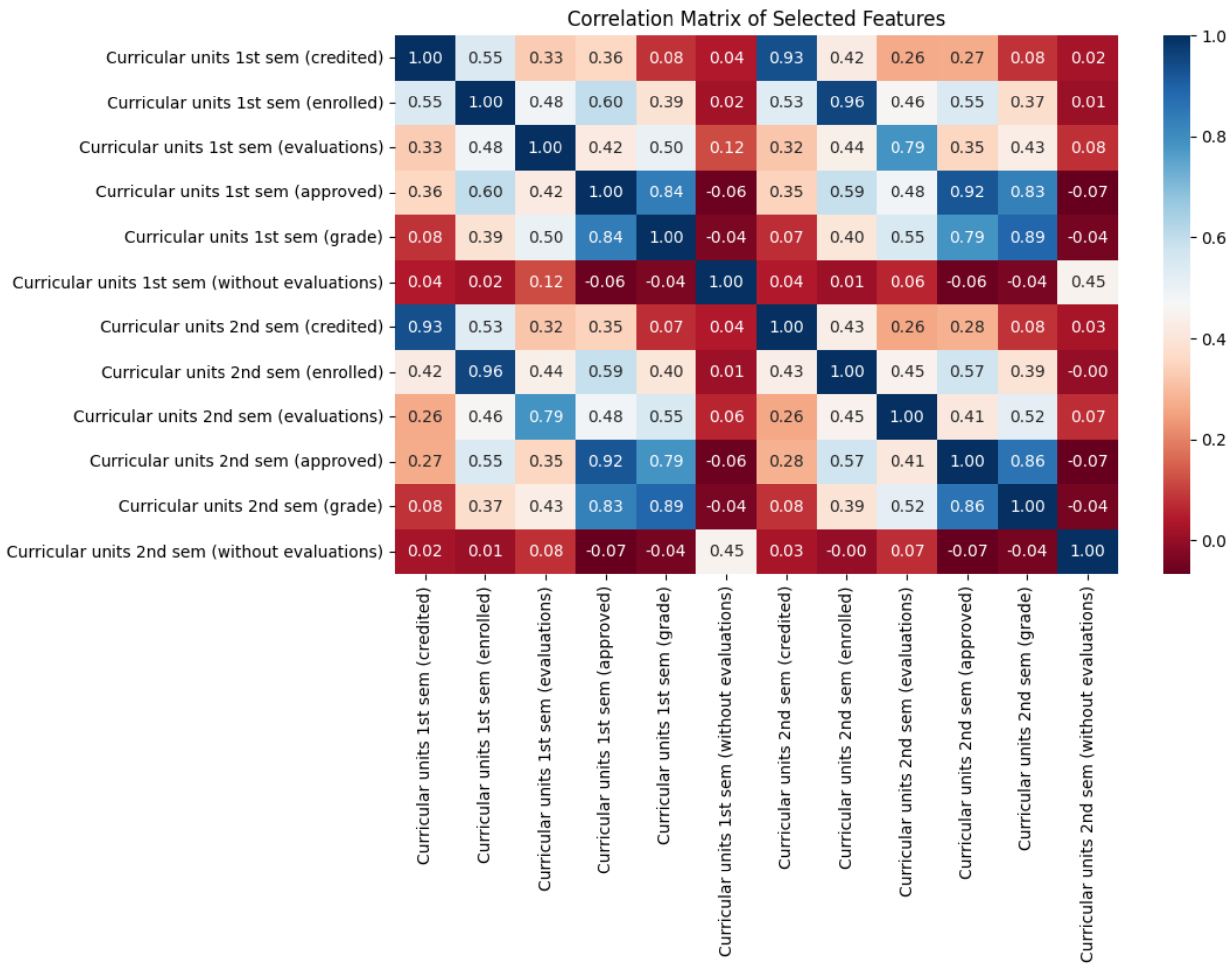


```python
custom_palette = {'Enrolled': '#bcbd22', 'Dropout': '#2ca02c',
'Graduate':'#00ff00'}

#plot violin chart for 1st and 2nd sem grade units vs Target
for column in ['Curricular units 1st sem (grade)', 'Curricular units
2nd sem (grade)']:
    sns.violinplot(data=train_data, x='Target', y=column,
hue='Target', palette=custom_palette, legend=False)
    plt.title(f'{column} vs Target')
    plt.show()
```


Curricular units 1st sem (grade) vs Target
Curricular units 1st sem (grade)
20
15
10
5
0
Graduate
Dropout
Enrolled
Target

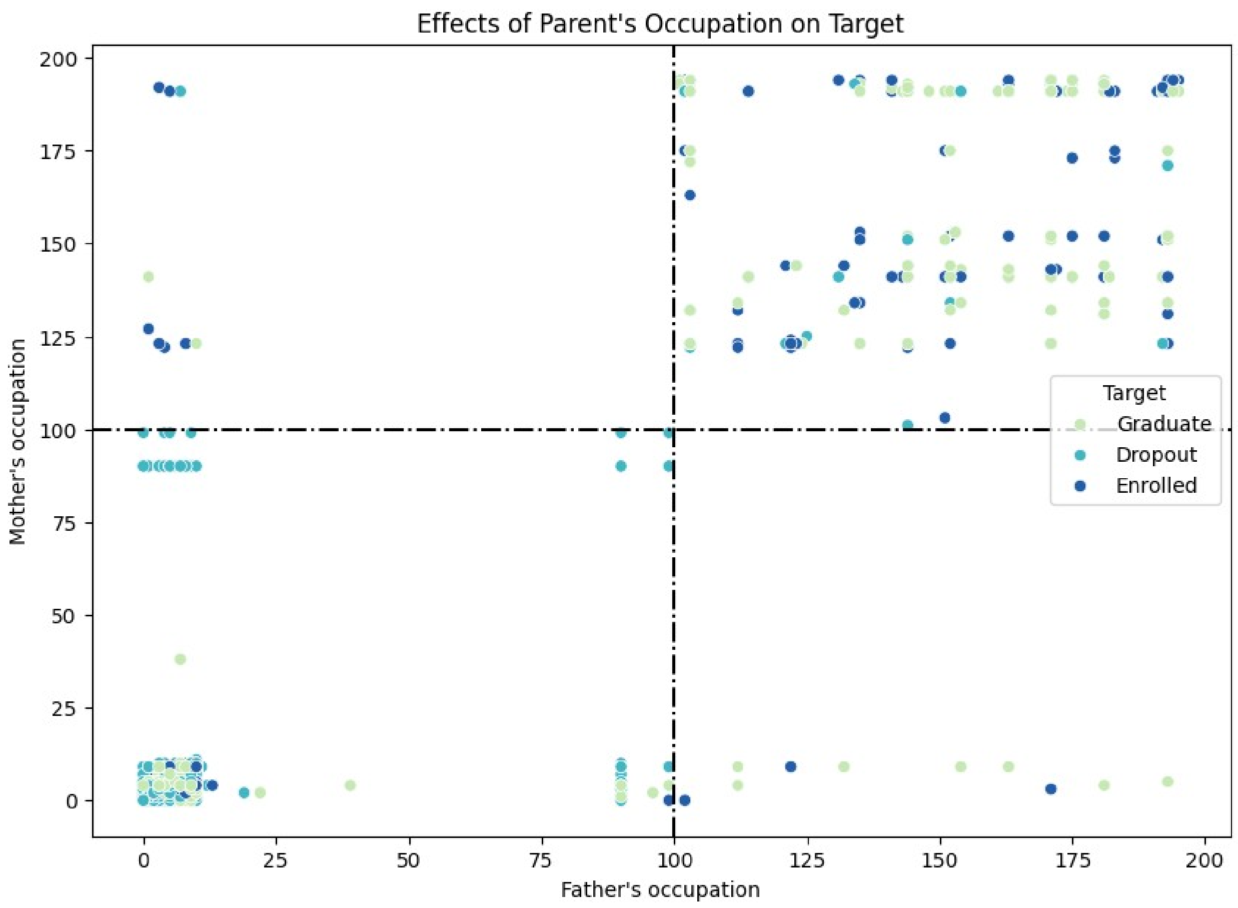


```
Columns_needed1 = ['Curricular units 1st sem (credited)', 'Curricular
units 1st sem (enrolled)', 'Curricular units 1st sem (evaluations)',
'Curricular units 1st sem (approved)', 'Curricular units 1st sem
(grade)', 'Curricular units 1st sem (without evaluations)',
'Curricular units 2nd sem (credited)', 'Curricular units 2nd sem
(enrolled)', 'Curricular units 2nd sem (evaluations)', 'Curricular
units 2nd sem (approved)', 'Curricular units 2nd sem (grade)',
'Curricular units 2nd sem (without evaluations)']
subset1 = train_data[Columns_needed1]

# Checking Correlation between the curricular units variables
correlation_matrix1 = subset1.corr()

plt.figure(figsize=(10, 6))
sns.heatmap(correlation_matrix1, annot=True, cmap= 'RdBu', fmt='.2f',
cbar=True)
plt.title("Correlation Matrix of Selected Features")
plt.show()
```

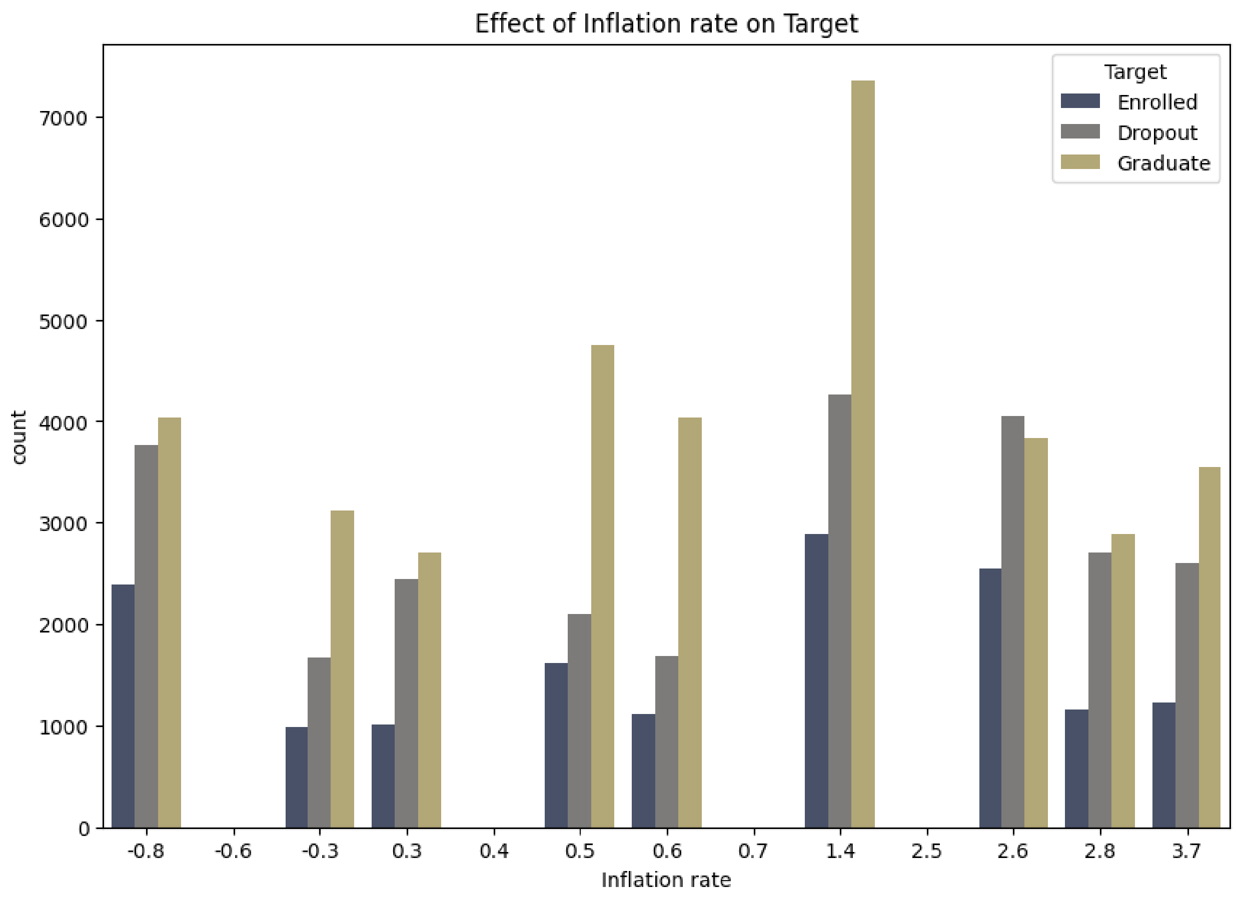


	1st credited	1st enrolled	1st evaluations	1st approved	1st grade	1st without eval	2nd credited	2nd enrolled	2nd evaluations	2nd approved	2nd grade	2nd without eval
Curricular units 1st sem (credited)	1.00	0.55	0.33	0.36	0.08	0.04	0.93	0.42	0.26	0.27	0.08	0.02
Curricular units 1st sem (enrolled)	0.55	1.00	0.48	0.60	0.39	0.02	0.53	0.96	0.46	0.55	0.37	0.01
Curricular units 1st sem (evaluations)	0.33	0.48	1.00	0.42	0.50	0.12	0.32	0.44	0.79	0.35	0.43	0.08
Curricular units 1st sem (approved)	0.36	0.60	0.42	1.00	0.84	-0.06	0.35	0.59	0.48	0.92	0.83	-0.07
Curricular units 1st sem (grade)	0.08	0.39	0.50	0.84	1.00	-0.04	0.07	0.40	0.55	0.79	0.89	-0.04
Curricular units 1st sem (without evaluations)	0.04	0.02	0.12	-0.06	-0.04	1.00	0.04	0.01	0.06	-0.06	-0.04	0.45
Curricular units 2nd sem (credited)	0.93	0.53	0.32	0.35	0.07	0.04	1.00	0.43	0.26	0.28	0.08	0.03
Curricular units 2nd sem (enrolled)	0.42	0.96	0.44	0.59	0.40	0.01	0.43	1.00	0.45	0.57	0.39	-0.00
Curricular units 2nd sem (evaluations)	0.26	0.46	0.79	0.48	0.55	0.06	0.26	0.45	1.00	0.41	0.52	0.07
Curricular units 2nd sem (approved)	0.27	0.55	0.35	0.92	0.79	-0.06	0.28	0.57	0.41	1.00	0.86	-0.07
Curricular units 2nd sem (grade)	0.08	0.37	0.43	0.83	0.89	-0.04	0.08	0.39	0.52	0.86	1.00	-0.04
Curricular units 2nd sem (without evaluations)	0.02	0.01	0.08	-0.07	-0.04	0.45	0.03	-0.00	0.07	-0.07	-0.04	1.00

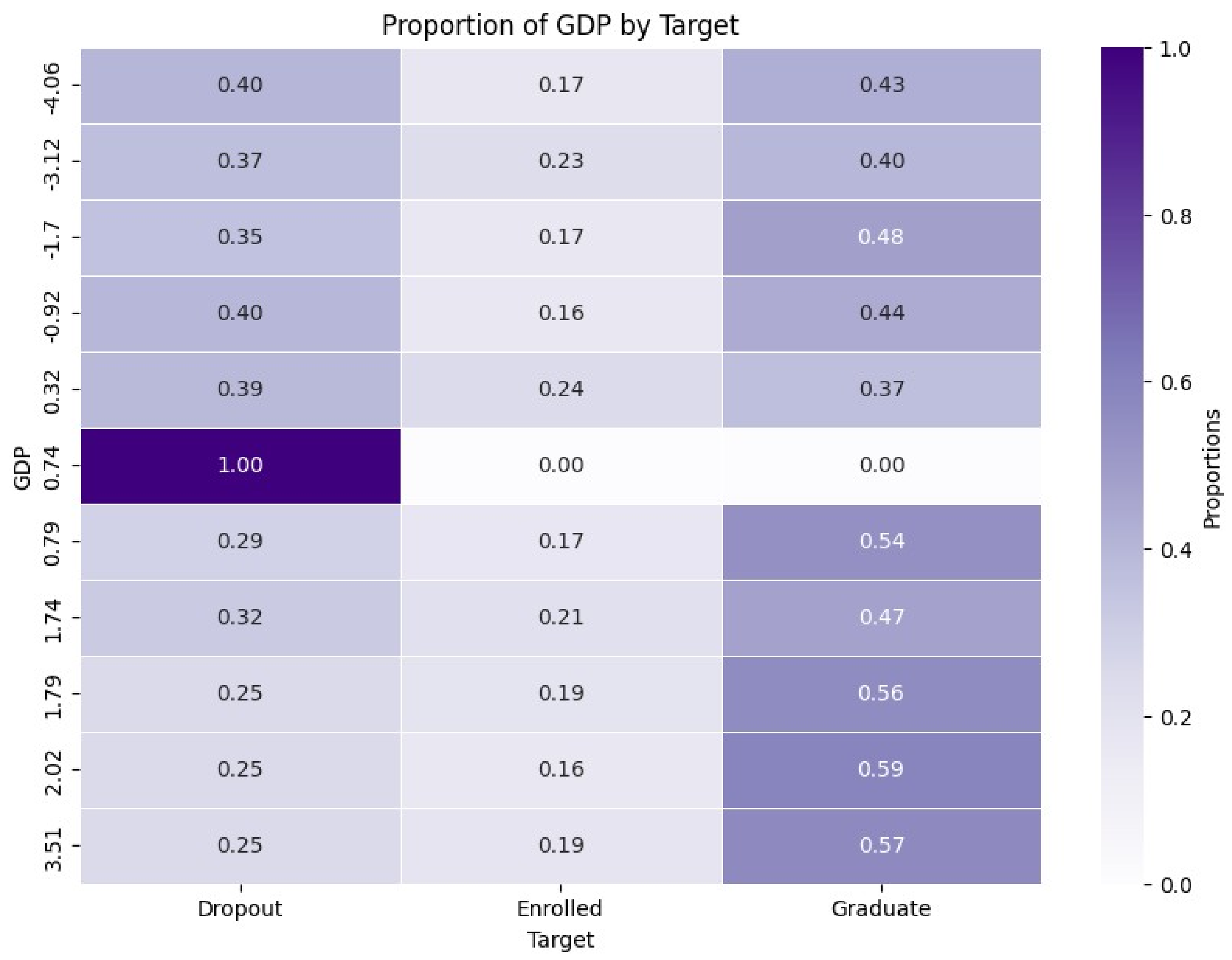
```python
# To check Parents occuption effects on Target

#fathers_target_counts = train.groupby(['Father\'s occupation',
'Target']).size().unstack(fill_value=0)
#fathers_target_counts.plot(kind='bar', stacked=True, figsize=(12, 6),
color=color_theme)

plt.figure(figsize=(10,7))

sns.scatterplot(x= train_data['Father\'s occupation'],
                y= train_data['Mother\'s occupation'],
                hue= train_data['Target'],
                palette='YlGnBu')
plt.title('Effects of Parent\'s Occupation on Target')
plt.axhline(y=100, linestyle='-.', color= 'black')
plt.axvline(x=100, linestyle='-.', color= 'black')
plt.show()
```



```
# Effect of Inflation rate on Target

plt.figure(figsize=(10,7))
sns.countplot(data= train_data, x= 'Inflation rate', hue= 'Target',
palette= 'cividis')
plt.title('Effect of Inflation rate on Target')
plt.show()
```



```
#Effect of GDP on Target

gdp_percent=
train_data.groupby(['GDP','Target']).size().unstack(fill_value=0)
gdp_percent= gdp_percent.div(gdp_percent.sum(axis=1), axis=0)

plt.figure(figsize=(10,7))
sns.heatmap(gdp_percent, annot=True, cmap='Purples', fmt='.2f',
linewidths=0.5, cbar_kws={'label': 'Proportions'})
plt.title('Proportion of GDP by Target')
plt.show()
```



GDP	Dropout	Enrolled	Graduate
-4.06	0.40	0.17	0.43
-3.12	0.37	0.23	0.40
-1.7	0.35	0.17	0.48
-0.92	0.40	0.16	0.44
0.32	0.39	0.24	0.37
0.74	1.00	0.00	0.00
0.79	0.29	0.17	0.54
1.74	0.32	0.21	0.47
1.79	0.25	0.19	0.56
2.02	0.25	0.16	0.59
3.51	0.25	0.19	0.57