Final Project

March 3, 2025

1 Predict Students' Dropout and Academic Success

1.1 Introduction

This dataset is put together by collecting the information about students enrolled in higher education in Europe between the academic years 2008/2009 to 2018/2019. It is acquired from several disjoint database. These include data from 17 undergraduate degrees from different fields of knowledge, such as agronomy, design, education, nursing, journalism, management, social service, and technologies. The dataset includes information known at the time of student enrollment (academic path, demographics, and social-economic factors) and the students' academic performance at the end of the first and second semesters. This dataset is found here https://archive.ics.uci.edu/dataset/697/predict+students+dropout+and+academic+success.

The main goal of this project is to predict the reasons that lead to student dropout/academic success. This helps educators, administrators, and institutions improve retention rates and student performance. By using data-driven approaches, educational institutions can identify at-risk students and intervene before problems become severe. Here's are some of the ways students can benefit:

1) Early Intervention: Institutions can intervene by offering additional support such as tutoring, counseling, or mentoring to students at risk of dropping out.

2) Customized Learning: Based on predictions, schools can tailor learning paths for students, offering more personalized education to those who need it most.

3) Resource Allocation: Predicting dropouts helps universities allocate resources more efficiently, ensuring that at-risk students get the right help.

4) Retention Strategies: Universities can use these predictions to create strategies and improve student retention. These might include improving student engagement, modifying teaching methods, or enhancing social support systems.

1.2 Data Understanding

[407]:
```
# importing required libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib.gridspec import GridSpec
from sklearn.naive_bayes import GaussianNB
```

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import accuracy_score, classification_report,␣
 ↪precision_score,precision_recall_fscore_support, roc_auc_score, log_loss,␣
 ↪confusion_matrix
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import plot_tree
from sklearn.ensemble import AdaBoostClassifier
from imblearn.combine import SMOTETomek
from imblearn.ensemble import EasyEnsembleClassifier
from sklearn.metrics import roc_curve, roc_auc_score
from sklearn.metrics import roc_curve, auc
from statsmodels.stats.outliers_influence import variance_inflation_factor
```

[326]:
```
import warnings
warnings.filterwarnings('ignore')
```

[327]:
```
df = pd.read_csv('data.csv')
df.head()
```

[327]:
```
  Marital status;Application mode;Application order;Course;"Daytime/evening
attendance\t";Previous qualification;Previous qualification
(grade);Nacionality;Mother's qualification;Father's qualification;Mother's
occupation;Father's occupation;Admission grade;Displaced;Educational special
needs;Debtor;Tuition fees up to date;Gender;Scholarship holder;Age at
enrollment;International;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);Unemployment rate;Inflation rate;GDP;Target
0  1;17;5;171;1;1;122.0;1;19;12;5;9;127.3;1;0;0;1…
1  1;15;1;9254;1;1;160.0;1;1;3;3;3;142.5;1;0;0;0;…
2  1;1;5;9070;1;1;122.0;1;37;37;9;9;124.8;1;0;0;0…
3  1;17;2;9773;1;1;122.0;1;38;37;5;3;119.6;1;0;0;…
4  2;39;1;8014;0;1;100.0;1;37;38;9;9;141.5;0;0;0;…
```

[328]:
```
#splitting data into rows and columns

feature_names = df.columns.tolist()
for line in feature_names:
```

```
        Type = line.split(";")
df[Type] = df.iloc[:, 0].str.split(';',expand=True)
df = df.iloc[:, 1:]
df.head()
```

[328]:

	Marital status	Application mode	Application order	Course	\
0	1	17	5	171	
1	1	15	1	9254	
2	1	1	5	9070	
3	1	17	2	9773	
4	2	39	1	8014	

	"Daytime/evening attendance\t"	Previous qualification	\
0	1	1	
1	1	1	
2	1	1	
3	1	1	
4	0	1	

	Previous qualification (grade)	Nacionality	Mother's qualification	\
0	122.0	1	19	
1	160.0	1	1	
2	122.0	1	37	
3	122.0	1	38	
4	100.0	1	37	

	Father's qualification	…	Curricular units 2nd sem (credited)	\
0	12	…	0	
1	3	…	0	
2	37	…	0	
3	37	…	0	
4	38	…	0	

	Curricular units 2nd sem (enrolled)	Curricular units 2nd sem (evaluations)	\
0	0	0	
1	6	6	
2	6	0	
3	6	10	
4	6	6	

	Curricular units 2nd sem (approved)	Curricular units 2nd sem (grade)	\
0	0	0.0	
1	6	13.666666666666666	
2	0	0.0	
3	5	12.4	
4	6	13.0	

```
  Curricular units 2nd sem (without evaluations) Unemployment rate  \
0                                                0              10.8
1                                                0              13.9
2                                                0              10.8
3                                                0               9.4
4                                                0              13.9

  Inflation rate    GDP    Target
0            1.4   1.74   Dropout
1           -0.3   0.79  Graduate
2            1.4   1.74   Dropout
3           -0.8  -3.12  Graduate
4           -0.3   0.79  Graduate

[5 rows x 37 columns]
```

[329]:
```
# Renaming some column names
df = df.rename(columns={'Nacionality': 'Nationality'})
df = df.rename(columns={'"Daytime/evening attendance\t"': 'Daytime/evening␣
 ↪attendance'})
```

[330]:
```
df.shape
```

[330]:
```
(4424, 37)
```

This dataset consists of 4424 student records and 37 features.

1.2.1 Description About Each Feature

Variable	Description
Marital status	1 – single 2 – married 3 – widower 4 – divorced 5 – facto union 6 – legally separated (Categorical)
Application mode	The method of application used by the student. (Categorical)
Application order	Application order (between 0 - first choice; and 9 last choice) (Numerical)

Variable	Description
Course	33 - Biofuel Production Technologies 171 - Animation and Multimedia Design 8014 - Social Service (evening attendance) 9003 - Agronomy 9070 - Communication Design 9085 - Veterinary Nursing 9119 - Informatics Engineering 9130 - Equinculture 9147 - Management 9238 - Social Service 9254 - Tourism 9500 - Nursing 9556 - Oral Hygiene 9670 - Advertising and Marketing Management 9773 - Journalism and Communication 9853 - Basic Education 9991 - Management (evening attendance). (Categorical)
Daytime/evening attendance	Whether the student attends classes during the day or in the evening. (Categorical)
Previous qualification	1 - Secondary education 2 - Higher education - bachelor's degree 3 - Higher education - degree 4 - Higher education - master's 5 - Higher education - doctorate 6 - Frequency of higher education 9 - 12th year of schooling - not completed 10 - 11th year of schooling - not completed 12 - Other - 11th year of schooling 14 - 10th year of schooling 15 - 10th year of schooling - not completed 19 - Basic education 3rd cycle (9th/10th/11th year) or equiv. 38 - Basic education 2nd cycle (6th/7th/8th year) or equiv. 39 - Technological specialization course 40 - Higher education - degree (1st cycle) 42 - Professional higher technical course 43 - Higher education - master (2nd cycle). (Categorical)
Previous qualification(grade)	The qualification obtained by the student before enrolling in higher education (between 0 and 200). (Numerical)

Variable	Description
Nationality	1 - Portuguese; 2 - German; 6 - Spanish; 11 - Italian; 13 - Dutch; 14 - English; 17 - Lithuanian; 21 - Angolan; 22 - Cape Verdean; 24 - Guinean; 25 - Mozambican; 26 - Santomean; 32 - Turkish; 41 - Brazilian; 62 - Romanian; 100 - Moldova (Republic of); 101 - Mexican; 103 - Ukrainian; 105 - Russian; 108 - Cuban; 109 - Colombian (Categorical)
Mother's qualification	The qualification of the student's mother. (Categorical)
Father's qualification	The qualification of the student's father. (Categorical)
Mother's occupation	The occupation of the student's mother. (Categorical)
Father's occupation	The occupation of the student's father. (Categorical)
Admission grade	Admission grade (between 0 and 200). (Continuous)
Displaced	A displaced student is a student who has enrolled in a different school or district than they were originally enrolled in due to a crisis 1 – yes 0 – no. (Categorical)
Educational special needs	Whether the student has any special educational needs. (Categorical)
Debtor	Whether the student is a debtor. (Categorical)
Tuition fees up to date	Whether the student's tuition fees are up to date. (Categorical)
Gender	The gender of the student. (Categorical)
Scholarship holder	Whether the student is a scholarship holder. (Categorical)
Age at enrollment	The age of the student at the time of enrollment. (Numerical)
International	Whether the student is an international student. (Categorical)
Curricular units 1st sem (credited)	Number of curricular units credited by the student in the first semester. (Numerical)
Curricular units 1st sem (enrolled)	Number of curricular units enrolled by the student in the first semester. (Numerical)
Curricular units 1st sem (evaluations)	Number of curricular units evaluated by the student in the first semester. (Numerical)
Curricular units 1st sem (approved)	Number of curricular units approved by the student in the first semester. (Numerical)

Variable	Description
Curricular units 1st sem (grade)	Grade average in the 1st semester (between 0 and 20)
Curricular units 1st sem (without evaluations)	Number of curricular units without evalutions in the 1st semester
Curricular units 2nd sem (credited)	Number of curricular units credited in the 2nd semester
Curricular units 2nd sem (enrolled)	Number of curricular units enrolled in the 2nd semester
Curricular units 2nd sem (evaluations)	Number of evaluations to curricular units in the 2nd semester
Curricular units 2nd sem (approved)	Number of curricular units approved in the 2nd semester
Curricular units 2nd sem (grade)	Grade average in the 2nd semester (between 0 and 20)
Curricular units 2nd sem (without evaluations)	Number of curricular units without evalutions in the 1st semester
Unemployment rate	Unemployment rate (%)
Inflation rate	Inflation rate (%)
GDP	GDP
Target	The problem is formulated as a three category classification task (dropout, enrolled, and graduate) at the end of the normal duration of the course

[331]:
```
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4424 entries, 0 to 4423
Data columns (total 37 columns):
 #   Column                                          Non-Null Count  Dtype
---  ------                                          --------------  -----
 0   Marital status                                  4424 non-null   object
 1   Application mode                                4424 non-null   object
 2   Application order                               4424 non-null   object
 3   Course                                          4424 non-null   object
 4   Daytime/evening attendance                      4424 non-null   object
 5   Previous qualification                          4424 non-null   object
 6   Previous qualification (grade)                  4424 non-null   object
 7   Nationality                                     4424 non-null   object
 8   Mother's qualification                          4424 non-null   object
 9   Father's qualification                          4424 non-null   object
 10  Mother's occupation                             4424 non-null   object
 11  Father's occupation                             4424 non-null   object
 12  Admission grade                                 4424 non-null   object
 13  Displaced                                       4424 non-null   object
 14  Educational special needs                       4424 non-null   object
```

```
 15  Debtor                                          4424 non-null   object
 16  Tuition fees up to date                         4424 non-null   object
 17  Gender                                          4424 non-null   object
 18  Scholarship holder                              4424 non-null   object
 19  Age at enrollment                               4424 non-null   object
 20  International                                   4424 non-null   object
 21  Curricular units 1st sem (credited)             4424 non-null   object
 22  Curricular units 1st sem (enrolled)             4424 non-null   object
 23  Curricular units 1st sem (evaluations)          4424 non-null   object
 24  Curricular units 1st sem (approved)             4424 non-null   object
 25  Curricular units 1st sem (grade)                4424 non-null   object
 26  Curricular units 1st sem (without evaluations)  4424 non-null   object
 27  Curricular units 2nd sem (credited)             4424 non-null   object
 28  Curricular units 2nd sem (enrolled)             4424 non-null   object
 29  Curricular units 2nd sem (evaluations)          4424 non-null   object
 30  Curricular units 2nd sem (approved)             4424 non-null   object
 31  Curricular units 2nd sem (grade)                4424 non-null   object
 32  Curricular units 2nd sem (without evaluations)  4424 non-null   object
 33  Unemployment rate                               4424 non-null   object
 34  Inflation rate                                  4424 non-null   object
 35  GDP                                             4424 non-null   object
 36  Target                                          4424 non-null   object
dtypes: object(37)
memory usage: 1.2+ MB
None
```

There are no missing values in this dataset. Lets convert to appropriate datatypes.

[332]:
```
#coverting to appropriate datatypes
df['Marital status'] = df['Marital status'].astype(int)
df['Application mode'] = df['Application mode'].astype(int)
df['Application order'] = df['Application order'].astype(int)
df['Course'] = df['Course'].astype(int)
df['Daytime/evening attendance'] = df['Daytime/evening attendance'].astype(int)
df['Previous qualification'] = df['Previous qualification'].astype(int)
df['Previous qualification (grade)'] = df['Previous qualification (grade)'].
 ↪astype(float)
df['Nationality'] = df['Nationality'].astype(int)
df["Mother's qualification"] = df["Mother's qualification"].astype(int)
df["Father's qualification"] = df["Father's qualification"].astype(int)
df["Mother's occupation"] = df["Mother's occupation"].astype(int)
df["Father's occupation"] = df["Father's occupation"].astype(int)
df['Admission grade'] = df['Admission grade'].astype(float)
df['Displaced'] = df['Displaced'].astype(int)
df['Educational special needs'] = df['Educational special needs'].astype(float)
df['Debtor'] = df['Debtor'].astype(str)
df['Tuition fees up to date'] = df['Tuition fees up to date'].astype(int)
df['Gender'] = df['Gender'].astype(int)
```

```
df['Scholarship holder'] = df['Scholarship holder'].astype(int)
df['Age at enrollment'] = df['Age at enrollment'].astype(int)
df['International'] = df['International'].astype(int)
df['Curricular units 1st sem (credited)'] = df['Curricular units 1st sem␣
 ↪(credited)'].astype(int)
df['Curricular units 1st sem (enrolled)'] = df['Curricular units 1st sem␣
 ↪(enrolled)'].astype(int)
df['Curricular units 1st sem (evaluations)'] = df['Curricular units 1st sem␣
 ↪(evaluations)'].astype(int)
df['Curricular units 1st sem (approved)'] = df['Curricular units 1st sem␣
 ↪(approved)'].astype(int)
df['Curricular units 1st sem (grade)'] = df['Curricular units 1st sem (grade)'].
 ↪astype(float)
df['Curricular units 1st sem (without evaluations)'] = df['Curricular units 1st␣
 ↪sem (without evaluations)'].astype(int)
df['Curricular units 2nd sem (credited)'] = df['Curricular units 2nd sem␣
 ↪(credited)'].astype(int)
df['Curricular units 2nd sem (enrolled)'] = df['Curricular units 2nd sem␣
 ↪(enrolled)'].astype(int)
df['Curricular units 2nd sem (evaluations)'] = df['Curricular units 2nd sem␣
 ↪(evaluations)'].astype(int)
df['Curricular units 2nd sem (approved)'] = df['Curricular units 2nd sem␣
 ↪(approved)'].astype(int)
df['Curricular units 2nd sem (grade)'] = df['Curricular units 2nd sem (grade)'].
 ↪astype(float)
df['Curricular units 2nd sem (without evaluations)'] = df['Curricular units 2nd␣
 ↪sem (without evaluations)'].astype(int)
df['Unemployment rate'] = df['Unemployment rate'].astype(float)
df['Inflation rate'] = df['Inflation rate'].astype(float)
df['GDP'] = df['GDP'].astype(float)
df['Target'] = df['Target'].astype(str)
```

[333]:
```
df.describe()
```

[333]:

	Marital status	Application mode	Application order	Course	\
count	4424.000000	4424.000000	4424.000000	4424.000000	
mean	1.178571	18.669078	1.727848	8856.642631	
std	0.605747	17.484682	1.313793	2063.566416	
min	1.000000	1.000000	0.000000	33.000000	
25%	1.000000	1.000000	1.000000	9085.000000	
50%	1.000000	17.000000	1.000000	9238.000000	
75%	1.000000	39.000000	2.000000	9556.000000	
max	6.000000	57.000000	9.000000	9991.000000	

	Daytime/evening attendance	Previous qualification	\
count	4424.000000	4424.000000	

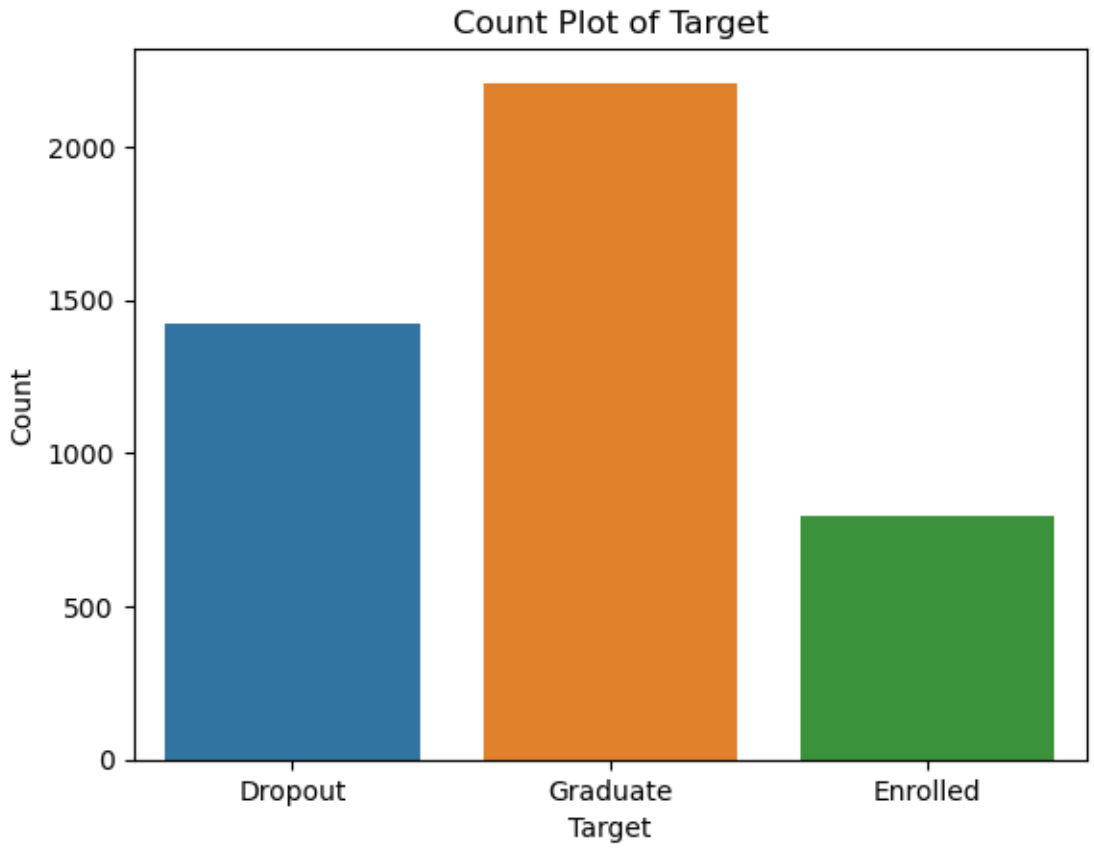
```
mean                     0.890823               4.577758
std                      0.311897              10.216592
min                      0.000000               1.000000
25%                      1.000000               1.000000
50%                      1.000000               1.000000
75%                      1.000000               1.000000
max                      1.000000              43.000000

       Previous qualification (grade)  Nationality  Mother's qualification  \
count                     4424.000000  4424.000000             4424.000000
mean                       132.613314     1.873192               19.561935
std                         13.188332     6.914514               15.603186
min                         95.000000     1.000000                1.000000
25%                        125.000000     1.000000                2.000000
50%                        133.100000     1.000000               19.000000
75%                        140.000000     1.000000               37.000000
max                        190.000000   109.000000               44.000000

       Father's qualification  …  \
count             4424.000000  …
mean                22.275316  …
std                 15.343108  …
min                  1.000000  …
25%                  3.000000  …
50%                 19.000000  …
75%                 37.000000  …
max                 44.000000  …

       Curricular units 1st sem (without evaluations)  \
count                                     4424.000000
mean                                        0.137658
std                                         0.690880
min                                         0.000000
25%                                         0.000000
50%                                         0.000000
75%                                         0.000000
max                                        12.000000

       Curricular units 2nd sem (credited)  \
count                           4424.000000
mean                               0.541817
std                                1.918546
min                                0.000000
25%                                0.000000
50%                                0.000000
75%                                0.000000
max                               19.000000
```

```
       Curricular units 2nd sem (enrolled)  \
count                          4424.000000
mean                              6.232143
std                               2.195951
min                               0.000000
25%                               5.000000
50%                               6.000000
75%                               7.000000
max                              23.000000

       Curricular units 2nd sem (evaluations)  \
count                               4424.000000
mean                                   8.063291
std                                    3.947951
min                                    0.000000
25%                                    6.000000
50%                                    8.000000
75%                                   10.000000
max                                   33.000000

       Curricular units 2nd sem (approved)  Curricular units 2nd sem (grade)  \
count                          4424.000000                       4424.000000
mean                              4.435805                         10.230206
std                               3.014764                          5.210808
min                               0.000000                          0.000000
25%                               2.000000                         10.750000
50%                               5.000000                         12.200000
75%                               6.000000                         13.333333
max                              20.000000                         18.571429

       Curricular units 2nd sem (without evaluations)  Unemployment rate  \
count                                     4424.000000        4424.000000
mean                                         0.150316          11.566139
std                                          0.753774           2.663850
min                                          0.000000           7.600000
25%                                          0.000000           9.400000
50%                                          0.000000          11.100000
75%                                          0.000000          13.900000
max                                         12.000000          16.200000

       Inflation rate          GDP
count     4424.000000  4424.000000
mean         1.228029     0.001969
std          1.382711     2.269935
min         -0.800000    -4.060000
25%          0.300000    -1.700000
```

```
50%          1.400000      0.320000
75%          2.600000      1.790000
max          3.700000      3.510000

[8 rows x 35 columns]
```

1.3 Data Exploring

[334]:
```
# Count plot of Target(Dropout, Graduate, Enrolled)
sns.countplot(x='Target', data=df)
plt.title('Count Plot of Target')
plt.xlabel('Target')
plt.ylabel('Count')
plt.show()
```



[335]:
```
# Calculate percentages
percentages = df['Target'].value_counts(normalize=True) * 100
print("\nTargetPercentages:\n", percentages)
```

```
TargetPercentages:
 Graduate    49.932188
Dropout     32.120253
Enrolled    17.947559
Name: Target, dtype: float64
```

Data Imbalance

There is a strong imbalance in target distribution among the classes. The majority class, Graduate, represents 50%, Dropout represents 32% and the minority class Enrolleg represents 18% of total records. This might result in a high prediction accuracy driven by the majority class at the expense of a poor performance of the minority class. So at the data-level approach, a sampling technique such as the Synthetic Minority Over Sampling Technique(SMOTE) is used to rebalance the target distribution.

[336]:
```
# Convert Target to numerical
df["Target"].replace('Dropout', 0, inplace=True)
df["Target"].replace('Graduate', 1, inplace=True)
df["Target"].replace('Enrolled', 2, inplace=True)
```

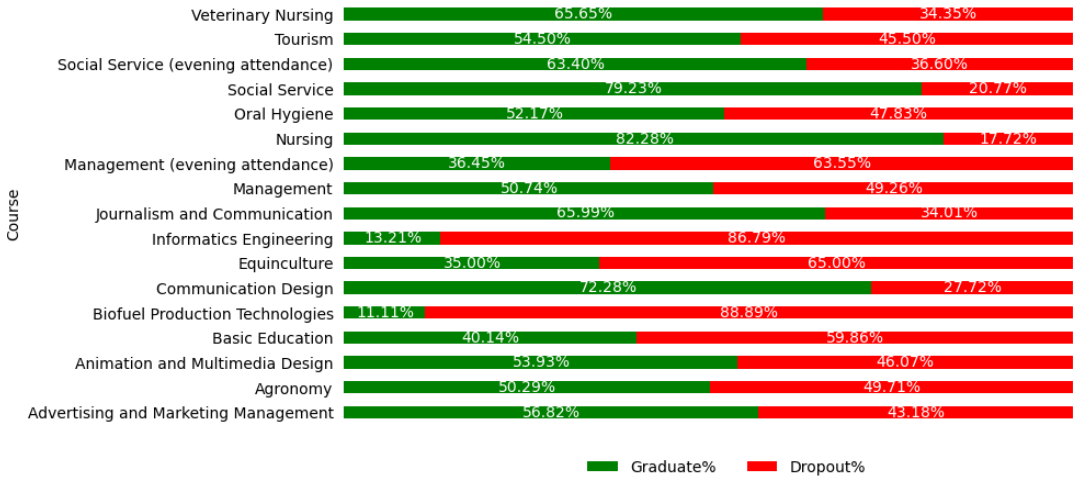
In the Target column, we are focussing on students who are “Dropout” or “Graduate”. So we are dropping those students who are enrolled.

[337]:
```
# Removing enrolled from the Target variable
df = df[df.Target != 2]
df.shape
```

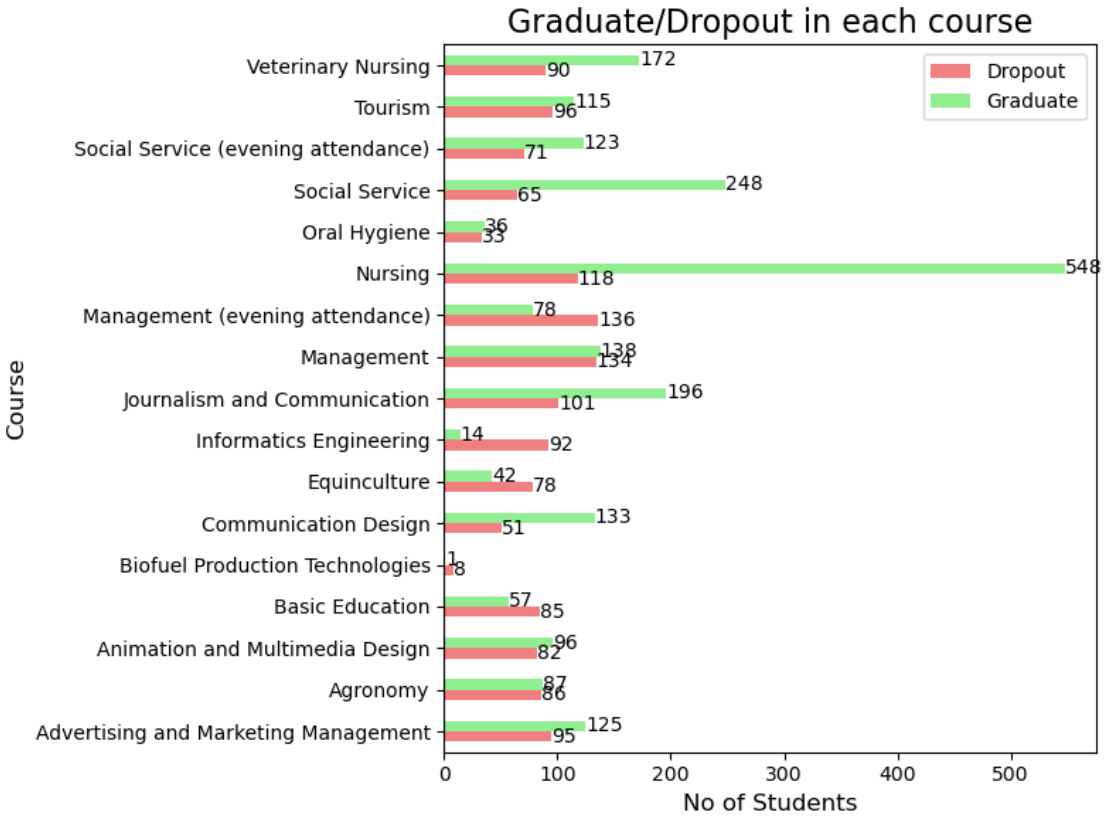
[337]:
```
(3630, 37)
```

[338]:
```
# Student outcome by course

df_Course = df.groupby(["Course", "Target"]).size().reset_index(name='Count')
mapping = {33: 'Biofuel Production Technologies', 171: 'Animation and␣
 ↪Multimedia Design',
           8014: 'Social Service (evening attendance)', 9003: 'Agronomy',
           9070: 'Communication Design',9085: 'Veterinary Nursing',
           9119: 'Informatics Engineering',9130: 'Equinculture',
           9147: 'Management',9238: 'Social Service',
           9254: 'Tourism',9500: 'Nursing',9556: 'Oral Hygiene',
           9670: 'Advertising and Marketing Management',9773: 'Journalism and␣
 ↪Communication',
           9853: 'Basic Education',9991: 'Management (evening attendance)'}

# Use the replace method to change values
df_Course['Course'] = df_Course['Course'].replace(mapping)

df_Course = df_Course.pivot(*df_Course).rename_axis(columns = None).
 ↪reset_index()
df_Course = df_Course.rename(columns={0: 'Dropout'})
```

```
df_Course = df_Course.rename(columns={1: 'Graduate'})
df_Course['Total'] = df_Course['Dropout']+df_Course['Graduate']
df_Course['Graduate%']=(df_Course['Graduate']/df_Course['Total']*100).round(2)
df_Course['Dropout%']=(df_Course['Dropout']/df_Course['Total']*100).round(2)

df_CoursePercent = df_Course
df_CoursePercent = df_CoursePercent.drop(['Dropout', 'Graduate', 'Total'],␣
 ↪axis=1)
df_CoursePercent = df_CoursePercent.set_index('Course')

# create the plot
ax = df_CoursePercent.plot(kind='barh', stacked=True, figsize=(9, 5),␣
 ↪color=['green', 'red'], xticks=[])
# move the legend
ax.legend(loc='upper center', bbox_to_anchor=(0.5, -0.05), ncol=2,␣
 ↪frameon=False)

# remove ticks
ax.tick_params(left=False, bottom=False)
# remove all spines
ax.spines[['top', 'bottom', 'left', 'right']].set_visible(False)

# iterate through each container
for c in ax.containers:

    # custom label calculates percent and add an empty string so 0 value bars␣
 ↪don't have a number
    labels = [f'{w:0.2f}%' if (w := v.get_width()) > 0 else '' for v in c]

    # add annotations
    ax.bar_label(c, labels=labels, label_type='center', padding=0.3, color='w')
```



[339]:
```
df_Course = df_Course.drop(['Dropout%', 'Graduate%', 'Total'], axis=1)
ax = df_Course.plot.barh(x='Course', color=['lightcoral', 'lightgreen'],␣
 ↪figsize=(8, 6))

# Adding count labels on top of the bars
for i in ax.containers:
    ax.bar_label(i, label_type='edge')

# Customizations
ax.set_title('Graduate/Dropout in each course', fontsize=16)
ax.set_xlabel('No of Students', fontsize=12)
ax.set_ylabel('Course', fontsize=12)
#ax.legend(title='Subcategories')

# Display the chart
plt.tight_layout()
plt.show()
```



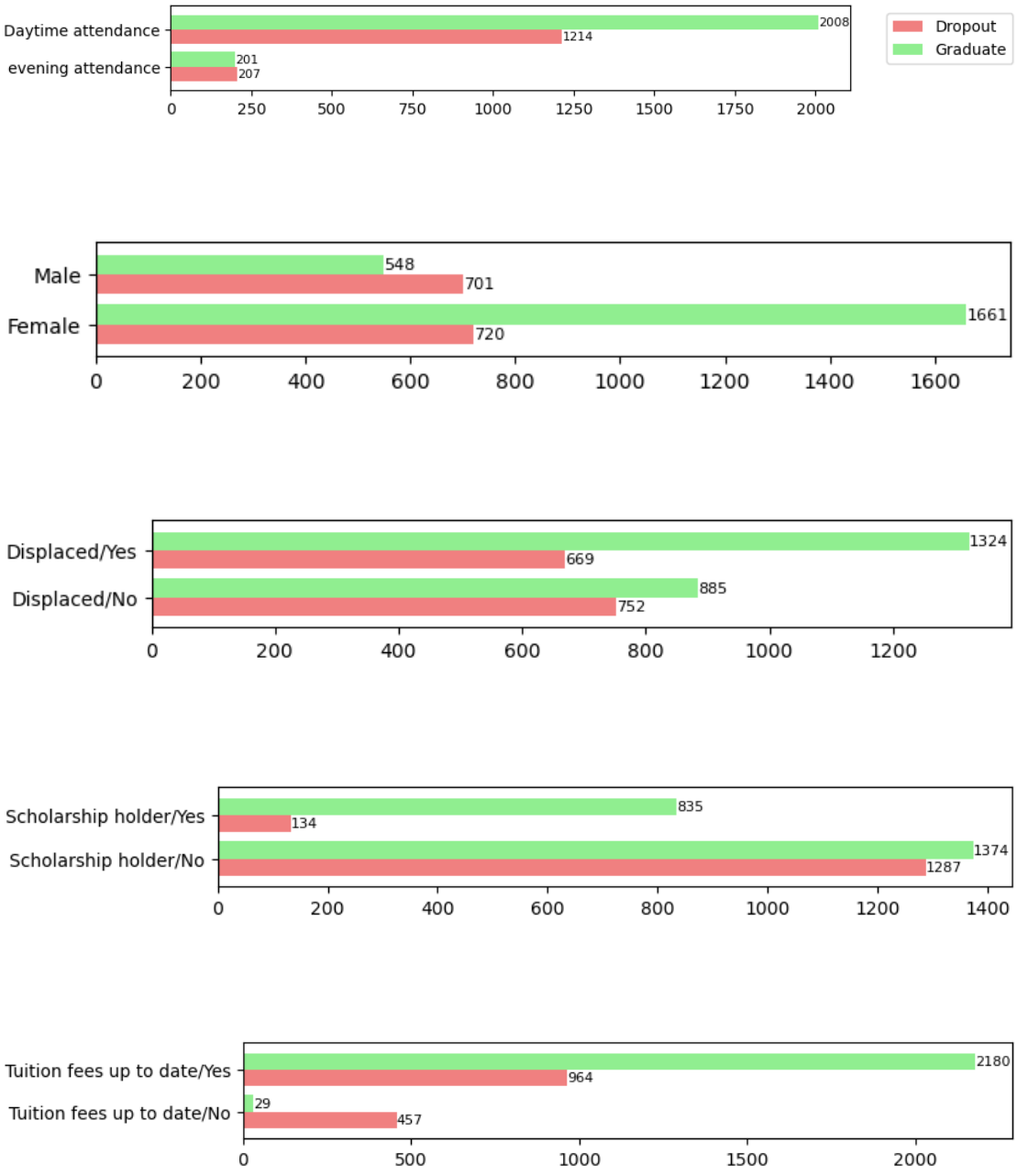
Most successful courses are Nursing and Social Service. On the opposite side, the technologies field with the courses of Biofuel Production Technologies and Informatics Engineering presents the most

unsuccessful results and the drop rate is also more.

[340]:
```
# Student outcome by gender, student displaced, tuition fees up to date,
# scholarship holder, and evening/daytime attendance.

df_studentoutcome = df[['Daytime/evening attendance', 'Displaced', 'Scholarship␣
 ↪holder',
                        'Gender','Tuition fees up to date','Target']].copy()
columns_to_plot = ['Daytime/evening attendance', 'Displaced',
                   'Scholarship holder','Gender','Tuition fees up to date']
df_studentoutcome["Target"].replace(1,'Graduate', inplace=True)
df_studentoutcome["Target"].replace(0, 'Dropout', inplace=True)



df_grouped =  df_studentoutcome.groupby(["Daytime/evening attendance",␣
 ↪"Target"]).size().reset_index(name='Count')
df_grouped = df_grouped.pivot(*df_grouped ).rename_axis(columns = None).
 ↪reset_index()
df_grouped["Daytime/evening attendance"].replace(0, 'evening attendance',␣
 ↪inplace=True)
df_grouped["Daytime/evening attendance"].replace(1,'Daytime attendance',␣
 ↪inplace=True)
ax1 = df_grouped.set_index('Daytime/evening attendance').plot.barh(figsize=(8,␣
 ↪1),width=0.8,color=['lightcoral', 'lightgreen'])

for i in ax1.containers:
    ax1.bar_label(i, label_type='edge',fontsize=8)
    #ax1.bar_label(i,color='w')
plt.ylabel("")
plt.legend(bbox_to_anchor=(1.25, 1), loc='upper right')


df_grouped =  df_studentoutcome.groupby(["Gender", "Target"]).size().
 ↪reset_index(name='Count')
df_grouped = df_grouped.pivot(*df_grouped ).rename_axis(columns = None).
 ↪reset_index()
df_grouped["Gender"].replace(0, 'Female', inplace=True)
df_grouped["Gender"].replace(1,'Male', inplace=True)
ax2 = df_grouped.set_index('Gender').plot.barh(figsize=(8, 1),width=0.
 ↪8,color=['lightcoral', 'lightgreen'])

for i in ax2.containers:
    ax2.bar_label(i, label_type='edge',fontsize=8)
ax2.get_legend().remove()
plt.ylabel("")
```

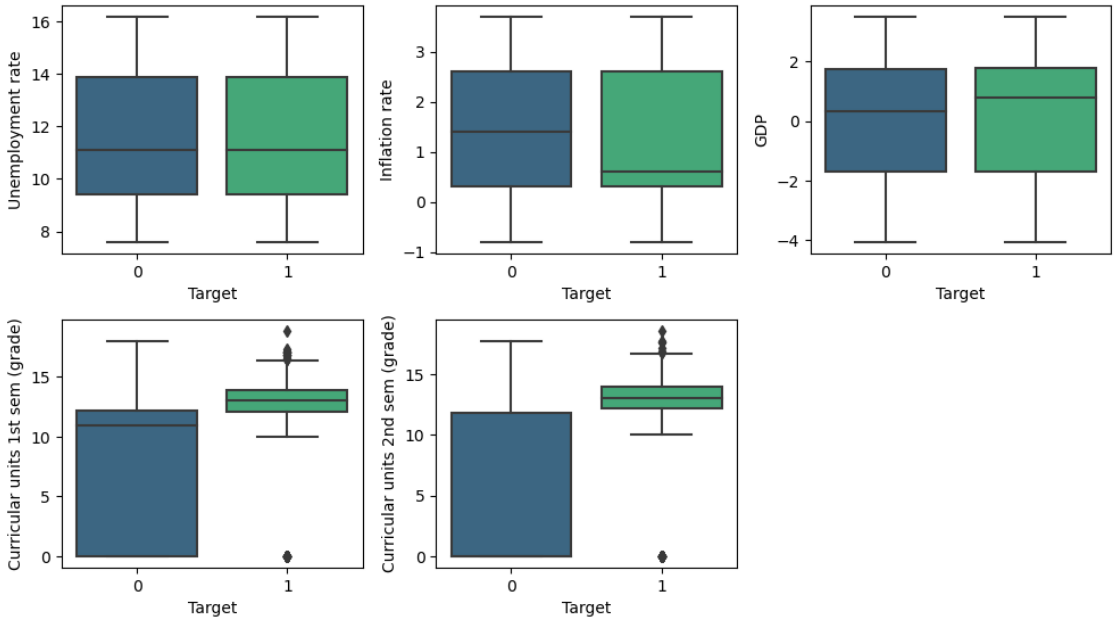
```
df_grouped =  df_studentoutcome.groupby(["Displaced", "Target"]).size().
 ↪reset_index(name='Count')
df_grouped = df_grouped.pivot(*df_grouped ).rename_axis(columns = None).
 ↪reset_index()
df_grouped["Displaced"].replace(0, 'Displaced/No', inplace=True)
df_grouped["Displaced"].replace(1,'Displaced/Yes', inplace=True)
ax3 = df_grouped.set_index('Displaced').plot.barh(figsize=(8, 1),width=0.
 ↪8,color=['lightcoral', 'lightgreen'])

for i in ax3.containers:
    ax3.bar_label(i, label_type='edge',fontsize=8)
ax3.get_legend().remove()
plt.ylabel("")


df_grouped =  df_studentoutcome.groupby(["Scholarship holder", "Target"]).
 ↪size().reset_index(name='Count')
df_grouped = df_grouped.pivot(*df_grouped ).rename_axis(columns = None).
 ↪reset_index()
df_grouped["Scholarship holder"].replace(0, 'Scholarship holder/No',␣
 ↪inplace=True)
df_grouped["Scholarship holder"].replace(1,'Scholarship holder/Yes',␣
 ↪inplace=True)
ax4 = df_grouped.set_index('Scholarship holder').plot.barh(figsize=(8,␣
 ↪1),width=0.8,color=['lightcoral', 'lightgreen'])

for i in ax4.containers:
    ax4.bar_label(i, label_type='edge',fontsize=8)
ax4.get_legend().remove()
plt.ylabel("")

df_grouped =  df_studentoutcome.groupby(["Tuition fees up to date", "Target"]).
 ↪size().reset_index(name='Count')
df_grouped = df_grouped.pivot(*df_grouped ).rename_axis(columns = None).
 ↪reset_index()
df_grouped["Tuition fees up to date"].replace(0, 'Tuition fees up to date/No',␣
 ↪inplace=True)
df_grouped["Tuition fees up to date"].replace(1,'Tuition fees up to date/Yes',␣
 ↪inplace=True)
ax5 = df_grouped.set_index('Tuition fees up to date').plot.barh(figsize=(8,␣
 ↪1),width=0.8,color=['lightcoral', 'lightgreen'])

for i in ax5.containers:
    ax5.bar_label(i, label_type='edge',fontsize=8)
```

```
ax5.get_legend().remove()
plt.ylabel("")

plt.show()
```



From this graph we can see that females are most successful, as well as the students that hold a scholarship and have their tuition fees up to date. Regarding the attendance regime (daytime or evening), the results show that students with daytime attendance finish the course earlier than evening students, as well as the students that are displaced from their homes.

[341]:
```
# Lets explore other continuous variables
df_continuousvar = df[['Unemployment rate', 'Inflation rate', 'GDP',
                       'Curricular units 1st sem (grade)','Curricular units␣
 ↪2nd sem (grade)','Target']].copy()
plt.figure(figsize=(10, 8))
gs = GridSpec(3, 3)

for i, col in enumerate(df_continuousvar.columns[:-1]):
    ax = plt.subplot(gs[i])
    sns.boxplot(x='Target', y=col, data=df_continuousvar, palette='viridis')
    ax.set_xlabel('Target', fontsize=10)
    ax.set_ylabel(col, fontsize=10)

plt.tight_layout()
plt.show()
```



1.4 Correlation

[342]:
```
correlation_matrix = df.corr()
correlation_matrix
```

[342]:

	Marital status
Marital status	1.000000
Application mode	0.274985
Application order	-0.130370
Course	0.042012

Daytime/evening attendance	-0.265823
Previous qualification	0.070984
Previous qualification (grade)	-0.038869
Nationality	-0.008899
Mother's qualification	0.185117
Father's qualification	0.124995
Mother's occupation	0.053892
Father's occupation	0.050499
Admission grade	-0.012440
Displaced	-0.240544
Educational special needs	-0.027434
Tuition fees up to date	-0.096920
Gender	-0.001124
Scholarship holder	-0.069684
Age at enrollment	0.522359
International	-0.026153
Curricular units 1st sem (credited)	0.066666
Curricular units 1st sem (enrolled)	0.058094
Curricular units 1st sem (evaluations)	0.059191
Curricular units 1st sem (approved)	-0.037691
Curricular units 1st sem (grade)	-0.069442
Curricular units 1st sem (without evaluations)	0.041535
Curricular units 2nd sem (credited)	0.067508
Curricular units 2nd sem (enrolled)	0.041256
Curricular units 2nd sem (evaluations)	0.030786
Curricular units 2nd sem (approved)	-0.058400
Curricular units 2nd sem (grade)	-0.079536
Curricular units 2nd sem (without evaluations)	0.025193
Unemployment rate	-0.018959
Inflation rate	0.011932
GDP	-0.028391
Target	-0.100479

	Application mode \
Marital status	0.274985
Application mode	1.000000
Application order	-0.287245
Course	0.063112
Daytime/evening attendance	-0.310854
Previous qualification	0.416666
Previous qualification (grade)	-0.071190
Nationality	-0.011711
Mother's qualification	0.122697
Father's qualification	0.098216
Mother's occupation	0.041650
Father's occupation	0.025629
Admission grade	-0.038311

Displaced	-0.308730
Educational special needs	-0.024611
Tuition fees up to date	-0.154143
Gender	0.186338
Scholarship holder	-0.174835
Age at enrollment	0.531601
International	-0.004620
Curricular units 1st sem (credited)	0.248883
Curricular units 1st sem (enrolled)	0.168857
Curricular units 1st sem (evaluations)	0.220351
Curricular units 1st sem (approved)	-0.037250
Curricular units 1st sem (grade)	-0.133919
Curricular units 1st sem (without evaluations)	0.054923
Curricular units 2nd sem (credited)	0.244575
Curricular units 2nd sem (enrolled)	0.130559
Curricular units 2nd sem (evaluations)	0.163626
Curricular units 2nd sem (approved)	-0.085270
Curricular units 2nd sem (grade)	-0.137288
Curricular units 2nd sem (without evaluations)	0.060434
Unemployment rate	0.072870
Inflation rate	-0.021895
GDP	-0.023945
Target	-0.244507

	Application order	Course	\
Marital status	-0.130370	0.042012	
Application mode	-0.287245	0.063112	
Application order	1.000000	0.056238	
Course	0.056238	1.000000	
Daytime/evening attendance	0.165494	-0.033887	
Previous qualification	-0.187881	-0.000230	
Previous qualification (grade)	-0.051715	-0.081792	
Nationality	-0.026706	-0.036492	
Mother's qualification	-0.058649	0.039976	
Father's qualification	-0.055254	0.043931	
Mother's occupation	-0.032459	0.021518	
Father's occupation	-0.025308	0.019156	
Admission grade	-0.092324	-0.123865	
Displaced	0.345791	-0.090136	
Educational special needs	0.025712	-0.019591	
Tuition fees up to date	0.061610	0.023489	
Gender	-0.106059	-0.094888	
Scholarship holder	0.072899	0.018123	
Age at enrollment	-0.281641	0.036323	
International	-0.030576	-0.024819	
Curricular units 1st sem (credited)	-0.133504	-0.081195	
Curricular units 1st sem (enrolled)	-0.015182	0.341673	

```
Curricular units 1st sem (evaluations)                -0.086004  0.264445
Curricular units 1st sem (approved)                    0.038274  0.188602
Curricular units 1st sem (grade)                       0.060776  0.381512
Curricular units 1st sem (without evaluations)        -0.038958  0.031539
Curricular units 2nd sem (credited)                   -0.127599 -0.078037
Curricular units 2nd sem (enrolled)                    0.033125  0.415262
Curricular units 2nd sem (evaluations)                -0.042519  0.281016
Curricular units 2nd sem (approved)                    0.072595  0.199739
Curricular units 2nd sem (grade)                       0.059817  0.335016
Curricular units 2nd sem (without evaluations)        -0.026822  0.033454
Unemployment rate                                     -0.099767  0.010375
Inflation rate                                        -0.004228  0.027855
GDP                                                    0.033031 -0.014411
Target                                                 0.094355  0.038135

                                                  Daytime/evening attendance  \
Marital status                                                      -0.265823
Application mode                                                    -0.310854
Application order                                                    0.165494
Course                                                              -0.033887
Daytime/evening attendance                                           1.000000
Previous qualification                                              -0.087616
Previous qualification (grade)                                       0.063810
Nationality                                                          0.030334
Mother's qualification                                              -0.188876
Father's qualification                                              -0.135725
Mother's occupation                                                 -0.034175
Father's occupation                                                 -0.030082
Admission grade                                                      0.018610
Displaced                                                            0.243653
Educational special needs                                            0.029207
Tuition fees up to date                                              0.052186
Gender                                                              -0.030507
Scholarship holder                                                   0.110240
Age at enrollment                                                   -0.453741
International                                                        0.032494
Curricular units 1st sem (credited)                                 -0.124038
Curricular units 1st sem (enrolled)                                 -0.041503
Curricular units 1st sem (evaluations)                              -0.048432
Curricular units 1st sem (approved)                                  0.028265
Curricular units 1st sem (grade)                                     0.073270
Curricular units 1st sem (without evaluations)                       0.039307
Curricular units 2nd sem (credited)                                 -0.105494
Curricular units 2nd sem (enrolled)                                  0.006915
Curricular units 2nd sem (evaluations)                               0.009977
Curricular units 2nd sem (approved)                                  0.054211
Curricular units 2nd sem (grade)                                     0.058371
```

Curricular units 2nd sem (without evaluations)	-0.010504
Unemployment rate	0.067192
Inflation rate	-0.017326
GDP	0.005007
Target	0.084496

	Previous qualification \
Marital status	0.070984
Application mode	0.416666
Application order	-0.187881
Course	-0.000230
Daytime/evening attendance	-0.087616
Previous qualification	1.000000
Previous qualification (grade)	0.089194
Nationality	-0.025211
Mother's qualification	-0.021399
Father's qualification	0.003865
Mother's occupation	0.009810
Father's occupation	0.015577
Admission grade	0.164962
Displaced	-0.126811
Educational special needs	-0.002446
Tuition fees up to date	-0.067020
Gender	0.094667
Scholarship holder	-0.071748
Age at enrollment	0.169046
International	-0.023995
Curricular units 1st sem (credited)	0.168644
Curricular units 1st sem (enrolled)	0.078829
Curricular units 1st sem (evaluations)	0.123573
Curricular units 1st sem (approved)	0.018125
Curricular units 1st sem (grade)	-0.010938
Curricular units 1st sem (without evaluations)	0.020155
Curricular units 2nd sem (credited)	0.147389
Curricular units 2nd sem (enrolled)	0.054375
Curricular units 2nd sem (evaluations)	0.094175
Curricular units 2nd sem (approved)	-0.010854
Curricular units 2nd sem (grade)	-0.008933
Curricular units 2nd sem (without evaluations)	0.022257
Unemployment rate	0.102451
Inflation rate	-0.070746
GDP	0.072121
Target	-0.062323

	Previous qualification (grade) \
Marital status	-0.038869

Application mode	-0.071190
Application order	-0.051715
Course	-0.081792
Daytime/evening attendance	0.063810
Previous qualification	0.089194
Previous qualification (grade)	1.000000
Nationality	0.053888
Mother's qualification	-0.073035
Father's qualification	-0.047147
Mother's occupation	-0.017383
Father's occupation	-0.022169
Admission grade	0.577241
Displaced	-0.010238
Educational special needs	0.009246
Tuition fees up to date	0.079246
Gender	-0.057654
Scholarship holder	0.069054
Age at enrollment	-0.133535
International	0.055494
Curricular units 1st sem (credited)	-0.017717
Curricular units 1st sem (enrolled)	-0.033503
Curricular units 1st sem (evaluations)	-0.072595
Curricular units 1st sem (approved)	0.054120
Curricular units 1st sem (grade)	0.077191
Curricular units 1st sem (without evaluations)	-0.011857
Curricular units 2nd sem (credited)	-0.026107
Curricular units 2nd sem (enrolled)	-0.036162
Curricular units 2nd sem (evaluations)	-0.061730
Curricular units 2nd sem (approved)	0.055232
Curricular units 2nd sem (grade)	0.068240
Curricular units 2nd sem (without evaluations)	-0.023654
Unemployment rate	0.043345
Inflation rate	0.019247
GDP	-0.054349
Target	0.109464

	Nationality \
Marital status	-0.008899
Application mode	-0.011711
Application order	-0.026706
Course	-0.036492
Daytime/evening attendance	0.030334
Previous qualification	-0.025211
Previous qualification (grade)	0.053888
Nationality	1.000000
Mother's qualification	-0.038168
Father's qualification	-0.073033

Mother's occupation	0.010331
Father's occupation	0.008297
Admission grade	0.013500
Displaced	-0.003521
Educational special needs	-0.003216
Tuition fees up to date	-0.035985
Gender	-0.024381
Scholarship holder	-0.002002
Age at enrollment	-0.004959
International	0.797387
Curricular units 1st sem (credited)	0.000178
Curricular units 1st sem (enrolled)	-0.020632
Curricular units 1st sem (evaluations)	-0.018671
Curricular units 1st sem (approved)	-0.005631
Curricular units 1st sem (grade)	-0.003854
Curricular units 1st sem (without evaluations)	-0.002042
Curricular units 2nd sem (credited)	-0.004657
Curricular units 2nd sem (enrolled)	-0.029476
Curricular units 2nd sem (evaluations)	-0.032890
Curricular units 2nd sem (approved)	-0.024523
Curricular units 2nd sem (grade)	-0.014859
Curricular units 2nd sem (without evaluations)	-0.014388
Unemployment rate	0.003110
Inflation rate	-0.005440
GDP	0.017080
Target	-0.015516

	Mother's qualification \
Marital status	0.185117
Application mode	0.122697
Application order	-0.058649
Course	0.039976
Daytime/evening attendance	-0.188876
Previous qualification	-0.021399
Previous qualification (grade)	-0.073035
Nationality	-0.038168
Mother's qualification	1.000000
Father's qualification	0.543890
Mother's occupation	0.093654
Father's occupation	0.067471
Admission grade	-0.068855
Displaced	-0.077115
Educational special needs	-0.011930
Tuition fees up to date	-0.031185
Gender	-0.042377
Scholarship holder	0.026869
Age at enrollment	0.291844

International	-0.020669
Curricular units 1st sem (credited)	0.048124
Curricular units 1st sem (enrolled)	0.051753
Curricular units 1st sem (evaluations)	0.050702
Curricular units 1st sem (approved)	-0.018028
Curricular units 1st sem (grade)	-0.044137
Curricular units 1st sem (without evaluations)	0.015112
Curricular units 2nd sem (credited)	0.041687
Curricular units 2nd sem (enrolled)	0.033219
Curricular units 2nd sem (evaluations)	0.033510
Curricular units 2nd sem (approved)	-0.026406
Curricular units 2nd sem (grade)	-0.034434
Curricular units 2nd sem (without evaluations)	0.029799
Unemployment rate	-0.122171
Inflation rate	0.057987
GDP	-0.076342
Target	-0.053989

	Father's qualification	…	\
Marital status	0.124995	…	
Application mode	0.098216	…	
Application order	-0.055254	…	
Course	0.043931	…	
Daytime/evening attendance	-0.135725	…	
Previous qualification	0.003865	…	
Previous qualification (grade)	-0.047147	…	
Nationality	-0.073033	…	
Mother's qualification	0.543890	…	
Father's qualification	1.000000	…	
Mother's occupation	0.070684	…	
Father's occupation	0.069976	…	
Admission grade	-0.057358	…	
Displaced	-0.066561	…	
Educational special needs	0.004516	…	
Tuition fees up to date	-0.024534	…	
Gender	-0.059075	…	
Scholarship holder	0.093947	…	
Age at enrollment	0.194840	…	
International	-0.069160	…	
Curricular units 1st sem (credited)	0.045511	…	
Curricular units 1st sem (enrolled)	0.044325	…	
Curricular units 1st sem (evaluations)	0.043276	…	
Curricular units 1st sem (approved)	0.008896	…	
Curricular units 1st sem (grade)	-0.006826	…	
Curricular units 1st sem (without evaluations)	-0.006681	…	
Curricular units 2nd sem (credited)	0.047034	…	
Curricular units 2nd sem (enrolled)	0.029756	…	

```
Curricular units 2nd sem (evaluations)                          0.014678  …
Curricular units 2nd sem (approved)                             0.001245  …
Curricular units 2nd sem (grade)                               -0.008768  …
Curricular units 2nd sem (without evaluations)                  0.004103  …
Unemployment rate                                              -0.075372  …
Inflation rate                                                  0.062772  …
GDP                                                            -0.059914  …
Target                                                         -0.005865  …

                                                    Curricular units 2nd sem
(credited)  \
Marital status
0.067508
Application mode
0.244575
Application order
-0.127599
Course
-0.078037
Daytime/evening attendance
-0.105494
Previous qualification
0.147389
Previous qualification (grade)
-0.026107
Nationality
-0.004657
Mother's qualification
0.041687
Father's qualification
0.047034
Mother's occupation
0.005302
Father's occupation
-0.009842
Admission grade
0.038345
Displaced
-0.096240
Educational special needs
-0.019928
Tuition fees up to date
0.020306
Gender
0.023027
Scholarship holder
-0.079378
```

```
Age at enrollment
0.196485
International
0.006237
Curricular units 1st sem (credited)
0.947093
Curricular units 1st sem (enrolled)
0.763276
Curricular units 1st sem (evaluations)
0.546980
Curricular units 1st sem (approved)
0.615834
Curricular units 1st sem (grade)
0.120199
Curricular units 1st sem (without evaluations)
0.144378
Curricular units 2nd sem (credited)
1.000000
Curricular units 2nd sem (enrolled)
0.683086
Curricular units 2nd sem (evaluations)
0.453716
Curricular units 2nd sem (approved)
0.522684
Curricular units 2nd sem (grade)
0.140181
Curricular units 2nd sem (without evaluations)
0.084506
Unemployment rate
0.013466
Inflation rate
0.019395
GDP
-0.038570
Target
0.052402

                                                    Curricular units 2nd sem
(enrolled)  \
Marital status
0.041256
Application mode
0.130559
Application order
0.033125
Course
0.415262
```

```
Daytime/evening attendance
0.006915
Previous qualification
0.054375
Previous qualification (grade)
-0.036162
Nationality
-0.029476
Mother's qualification
0.033219
Father's qualification
0.029756
Mother's occupation
-0.020567
Father's occupation
-0.032043
Admission grade
-0.040766
Displaced
-0.046431
Educational special needs
-0.032488
Tuition fees up to date
0.103347
Gender
-0.126659
Scholarship holder
0.024982
Age at enrollment
0.071177
International
-0.019606
Curricular units 1st sem (credited)
0.650707
Curricular units 1st sem (enrolled)
0.941286
Curricular units 1st sem (evaluations)
0.625967
Curricular units 1st sem (approved)
0.737375
Curricular units 1st sem (grade)
0.407084
Curricular units 1st sem (without evaluations)
0.123588
Curricular units 2nd sem (credited)
0.683086
Curricular units 2nd sem (enrolled)
```

```
1.000000
Curricular units 2nd sem (evaluations)
0.625080
Curricular units 2nd sem (approved)
0.704445
Curricular units 2nd sem (grade)
0.400773
Curricular units 2nd sem (without evaluations)
0.070363
Unemployment rate
0.066219
Inflation rate
0.028390
GDP
-0.026076
Target
0.182897

                                                    Curricular units 2nd sem
(evaluations)  \
Marital status
0.030786
Application mode
0.163626
Application order
-0.042519
Course
0.281016
Daytime/evening attendance
0.009977
Previous qualification
0.094175
Previous qualification (grade)
-0.061730
Nationality
-0.032890
Mother's qualification
0.033510
Father's qualification
0.014678
Mother's occupation
-0.026873
Father's occupation
-0.033926
Admission grade
-0.060954
Displaced
```

-0.035874
Educational special needs
-0.022319
Tuition fees up to date
0.055691
Gender
-0.048000
Scholarship holder
0.005903
Age at enrollment
0.059554
International
-0.013539
Curricular units 1st sem (credited)
0.448136
Curricular units 1st sem (enrolled)
0.618506
Curricular units 1st sem (evaluations)
0.790616
Curricular units 1st sem (approved)
0.579089
Curricular units 1st sem (grade)
0.503365
Curricular units 1st sem (without evaluations)
0.170004
Curricular units 2nd sem (credited)
0.453716
Curricular units 2nd sem (enrolled)
0.625080
Curricular units 2nd sem (evaluations)
1.000000
Curricular units 2nd sem (approved)
0.508968
Curricular units 2nd sem (grade)
0.463194
Curricular units 2nd sem (without evaluations)
0.166443
Unemployment rate
0.055229
Inflation rate
-0.001422
GDP
-0.021846
Target
0.119239

Curricular units 2nd sem

```
(approved)  \
Marital status
-0.058400
Application mode
-0.085270
Application order
0.072595
Course
0.199739
Daytime/evening attendance
0.054211
Previous qualification
-0.010854
Previous qualification (grade)
0.055232
Nationality
-0.024523
Mother's qualification
-0.026406
Father's qualification
0.001245
Mother's occupation
-0.031209
Father's occupation
-0.030994
Admission grade
0.089429
Displaced
0.076466
Educational special needs
-0.018142
Tuition fees up to date
0.329017
Gender
-0.234663
Scholarship holder
0.214997
Age at enrollment
-0.147668
International
-0.010648
Curricular units 1st sem (credited)
0.495762
Curricular units 1st sem (enrolled)
0.674880
Curricular units 1st sem (evaluations)
0.466744
```

```
Curricular units 1st sem (approved)
0.916334
Curricular units 1st sem (grade)
0.691907
Curricular units 1st sem (without evaluations)
0.001458
Curricular units 2nd sem (credited)
0.522684
Curricular units 2nd sem (enrolled)
0.704445
Curricular units 2nd sem (evaluations)
0.508968
Curricular units 2nd sem (approved)
1.000000
Curricular units 2nd sem (grade)
0.786838
Curricular units 2nd sem (without evaluations)
-0.052389
Unemployment rate
0.040061
Inflation rate
-0.026751
GDP
0.014514
Target
0.653995
```

	Curricular units 2nd sem (grade)
\	
Marital status	-0.079536
Application mode	-0.137288
Application order	0.059817
Course	0.335016
Daytime/evening attendance	0.058371
Previous qualification	-0.008933
Previous qualification (grade)	0.068240
Nationality	-0.014859
Mother's qualification	-0.034434
Father's qualification	-0.008768
Mother's occupation	-0.030782
Father's occupation	-0.025479
Admission grade	0.095342
Displaced	0.084463
Educational special needs	-0.014808
Tuition fees up to date	0.318721
Gender	-0.219696
Scholarship holder	0.212342

```
Age at enrollment                                               -0.194145
International                                                   -0.003066
Curricular units 1st sem (credited)                              0.142343
Curricular units 1st sem (enrolled)                              0.366015
Curricular units 1st sem (evaluations)                           0.353550
Curricular units 1st sem (approved)                              0.709368
Curricular units 1st sem (grade)                                 0.845864
Curricular units 1st sem (without evaluations)                  -0.068725
Curricular units 2nd sem (credited)                              0.140181
Curricular units 2nd sem (enrolled)                              0.400773
Curricular units 2nd sem (evaluations)                           0.463194
Curricular units 2nd sem (approved)                              0.786838
Curricular units 2nd sem (grade)                                 1.000000
Curricular units 2nd sem (without evaluations)                  -0.079508
Unemployment rate                                                0.000261
Inflation rate                                                  -0.042639
GDP                                                              0.066077
Target                                                           0.605350

                                                 Curricular units 2nd sem
(without evaluations)  \
Marital status
0.025193
Application mode
0.060434
Application order
-0.026822
Course
0.033454
Daytime/evening attendance
-0.010504
Previous qualification
0.022257
Previous qualification (grade)
-0.023654
Nationality
-0.014388
Mother's qualification
0.029799
Father's qualification
0.004103
Mother's occupation
0.028571
Father's occupation
-0.000395
Admission grade
-0.012823
```

```
Displaced
-0.041696
Educational special needs
-0.013011
Tuition fees up to date
-0.095139
Gender
0.057755
Scholarship holder
-0.060597
Age at enrollment
0.080135
International
-0.012660
Curricular units 1st sem (credited)
0.069001
Curricular units 1st sem (enrolled)
0.075677
Curricular units 1st sem (evaluations)
0.158399
Curricular units 1st sem (approved)
-0.038974
Curricular units 1st sem (grade)
-0.055861
Curricular units 1st sem (without evaluations)
0.601573
Curricular units 2nd sem (credited)
0.084506
Curricular units 2nd sem (enrolled)
0.070363
Curricular units 2nd sem (evaluations)
0.166443
Curricular units 2nd sem (approved)
-0.052389
Curricular units 2nd sem (grade)
-0.079508
Curricular units 2nd sem (without evaluations)
1.000000
Unemployment rate
0.004067
Inflation rate
-0.030552
GDP
-0.079857
Target
-0.102687
```

	Unemployment rate
Marital status	-0.018959
Application mode	0.072870
Application order	-0.099767
Course	0.010375
Daytime/evening attendance	0.067192
Previous qualification	0.102451
Previous qualification (grade)	0.043345
Nationality	0.003110
Mother's qualification	-0.122171
Father's qualification	-0.075372
Mother's occupation	-0.110053
Father's occupation	-0.120888
Admission grade	0.037429
Displaced	-0.120367
Educational special needs	0.043913
Tuition fees up to date	0.009479
Gender	0.030020
Scholarship holder	0.066351
Age at enrollment	0.027129
International	-0.008089
Curricular units 1st sem (credited)	0.013313
Curricular units 1st sem (enrolled)	0.039788
Curricular units 1st sem (evaluations)	0.067011
Curricular units 1st sem (approved)	0.042251
Curricular units 1st sem (grade)	0.010272
Curricular units 1st sem (without evaluations)	-0.035046
Curricular units 2nd sem (credited)	0.013466
Curricular units 2nd sem (enrolled)	0.066219
Curricular units 2nd sem (evaluations)	0.055229
Curricular units 2nd sem (approved)	0.040061
Curricular units 2nd sem (grade)	0.000261
Curricular units 2nd sem (without evaluations)	0.004067
Unemployment rate	1.000000
Inflation rate	-0.029666
GDP	-0.341742
Target	0.004198

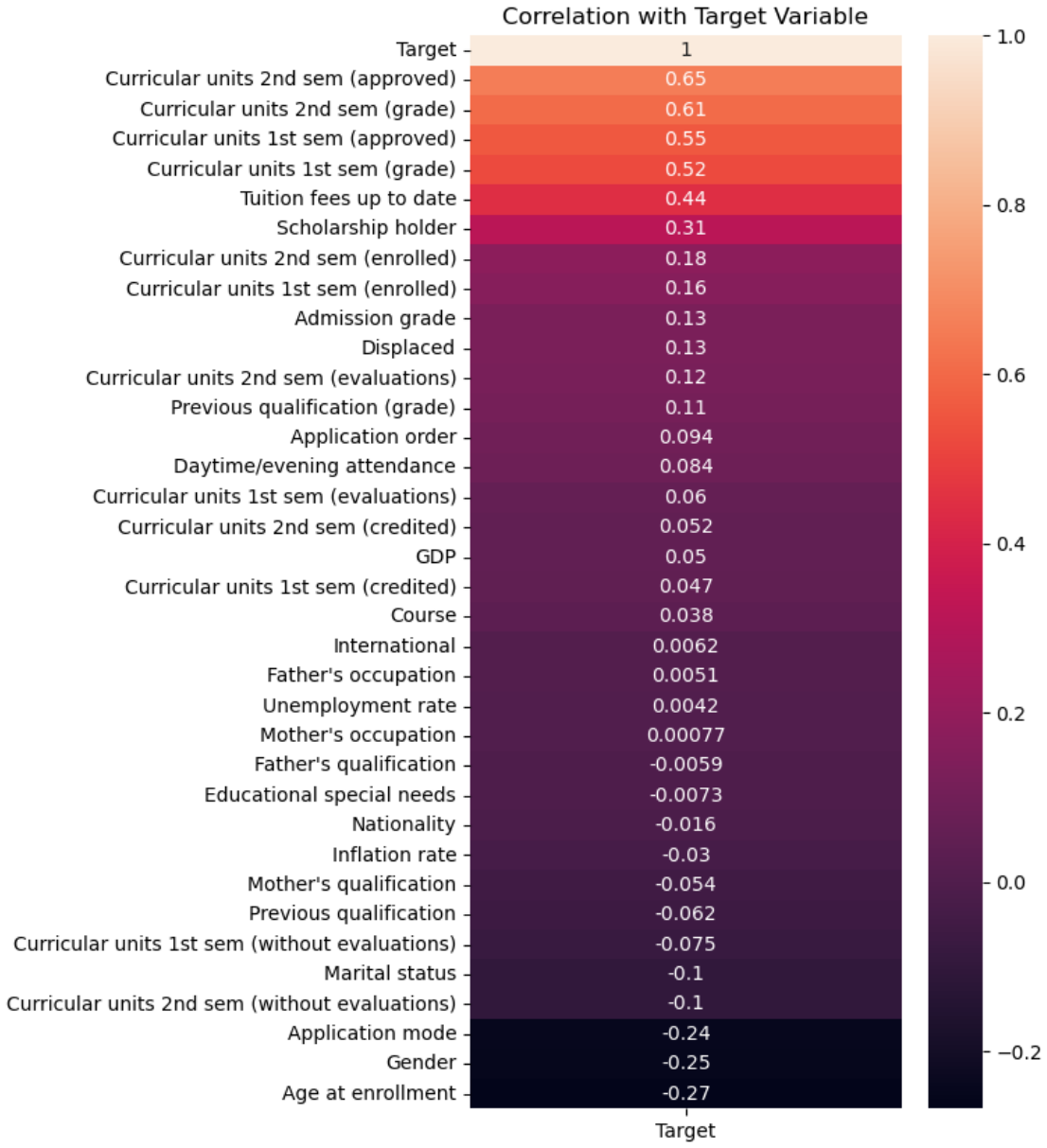
	Inflation rate	GDP
Marital status	0.011932	-0.028391
Application mode	-0.021895	-0.023945
Application order	-0.004228	0.033031
Course	0.027855	-0.014411
Daytime/evening attendance	-0.017326	0.005007
Previous qualification	-0.070746	0.072121
Previous qualification (grade)	0.019247	-0.054349
Nationality	-0.005440	0.017080

Mother's qualification	0.057987	-0.076342
Father's qualification	0.062772	-0.059914
Mother's occupation	0.051366	0.093200
Father's occupation	0.058573	0.102032
Admission grade	-0.023619	-0.014624
Displaced	-0.009026	0.064923
Educational special needs	0.004452	0.010978
Tuition fees up to date	-0.011067	0.027771
Gender	-0.004466	-0.021931
Scholarship holder	-0.044622	0.037842
Age at enrollment	0.030085	-0.069184
International	-0.005909	0.023997
Curricular units 1st sem (credited)	0.030038	-0.044169
Curricular units 1st sem (enrolled)	0.051167	-0.047367
Curricular units 1st sem (evaluations)	0.008854	-0.111698
Curricular units 1st sem (approved)	-0.003858	-0.000120
Curricular units 1st sem (grade)	-0.029068	0.054168
Curricular units 1st sem (without evaluations)	-0.037235	-0.130766
Curricular units 2nd sem (credited)	0.019395	-0.038570
Curricular units 2nd sem (enrolled)	0.028390	-0.026076
Curricular units 2nd sem (evaluations)	-0.001422	-0.021846
Curricular units 2nd sem (approved)	-0.026751	0.014514
Curricular units 2nd sem (grade)	-0.042639	0.066077
Curricular units 2nd sem (without evaluations)	-0.030552	-0.079857
Unemployment rate	-0.029666	-0.341742
Inflation rate	1.000000	-0.125789
GDP	-0.125789	1.000000
Target	-0.030326	0.050260

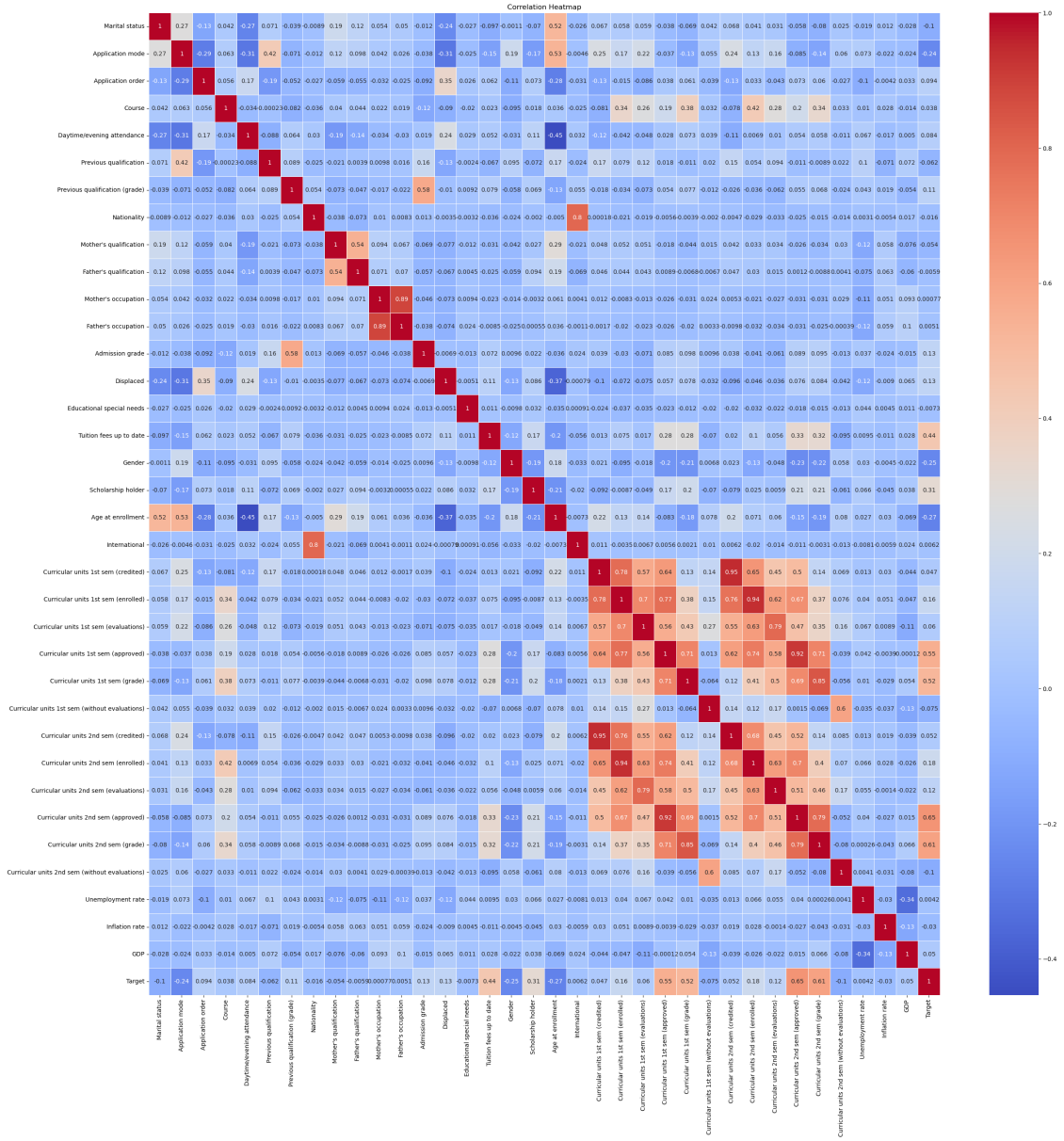
	Target
Marital status	-0.100479
Application mode	-0.244507
Application order	0.094355
Course	0.038135
Daytime/evening attendance	0.084496
Previous qualification	-0.062323
Previous qualification (grade)	0.109464
Nationality	-0.015516
Mother's qualification	-0.053989
Father's qualification	-0.005865
Mother's occupation	0.000772
Father's occupation	0.005066
Admission grade	0.128058
Displaced	0.126113
Educational special needs	-0.007254
Tuition fees up to date	0.442138
Gender	-0.251955

```
Scholarship holder                                0.313018
Age at enrollment                                -0.267229
International                                     0.006181
Curricular units 1st sem (credited)               0.046900
Curricular units 1st sem (enrolled)               0.161074
Curricular units 1st sem (evaluations)            0.059786
Curricular units 1st sem (approved)               0.554881
Curricular units 1st sem (grade)                  0.519927
Curricular units 1st sem (without evaluations) -0.074642
Curricular units 2nd sem (credited)               0.052402
Curricular units 2nd sem (enrolled)               0.182897
Curricular units 2nd sem (evaluations)            0.119239
Curricular units 2nd sem (approved)               0.653995
Curricular units 2nd sem (grade)                  0.605350
Curricular units 2nd sem (without evaluations) -0.102687
Unemployment rate                                 0.004198
Inflation rate                                   -0.030326
GDP                                               0.050260
Target                                            1.000000

[36 rows x 36 columns]
```

[343]:
```
# Correlation between Target and other features
plt.figure(figsize = (5,10))
sns.heatmap(df.corr()[['Target']].sort_values(by='Target', ascending=False),␣
 ↪annot = True)
plt.title('Correlation with Target Variable')
plt.show()
```



[344]:
```
plt.figure(figsize=(30, 30))
sns.heatmap(df.corr() , annot=True, cmap='coolwarm', linewidths=0.5)
plt.title('Correlation Heatmap')
plt.show()
```



Multi-collinearity There are some pairs of features having high correlation coefficients, which increases multi-collinearity in the dataset. The graph shows that the correlation is strongest in features in the same groups, such as "Nationality" and "International" or "Mother's occupation" and "Father's occupation", but also between the groups related with the performance at the end of the first semester and the second semester, such as "Curricular units 1st sem (approved)" and "Curricular units 2nd sem (approved)". To detect multi-collinearity variance inflation factor (VIF) score is calculated.

VIF score interpretation

VIF = 1: No correlation between variables

VIF between 1 and 5: Moderate correlation

VIF greater than 5: High correlation

VIF greater than 10: Serious correlation that may require further investigation

[345]:
```
#creating df_columns dataset
df_columns = df[['Unemployment rate', 'Inflation rate', 'GDP',
                 'Curricular units 1st sem (grade)','Curricular units 2nd sem␣
 ↪(grade)',
                 'Curricular units 1st sem (credited)','Curricular units 1st␣
 ↪sem (enrolled)',
                 'Curricular units 1st sem (evaluations)','Curricular units 1st␣
 ↪sem (without evaluations)',
                 'Curricular units 1st sem (approved)',
                 'Curricular units 2nd sem (credited)','Curricular units 2nd␣
 ↪sem (enrolled)',
                 'Curricular units 2nd sem (evaluations)','Curricular units 2nd␣
 ↪sem (approved)',
                 'Curricular units 2nd sem (without evaluations)',
                 'Nationality',"Mother's qualification", "Father's␣
 ↪qualification",
                 "Mother's occupation", "Father's occupation", 'Admission␣
 ↪grade',
                 'Application mode', 'Application order',␣
 ↪'Course','International']].copy()
```

[346]:
```
#calculate VIF

def calc_vif(X):

    # Calculating VIF
    vif = pd.DataFrame()
    vif["variables"] = X.columns
    #print(X)
    #print(X.values)
    for i in range(X.shape[1]):
        #print(i)
        #print(variance_inflation_factor(X.values, i))
        vif["VIF"] = [variance_inflation_factor(X.values, i) for i in range(X.
 ↪shape[1])]
    #print(vif)
    return(vif)
```

[347]:
```
X = df.iloc[:,:-1]
calc_vif(df_columns)
```

[347]:

	variables	VIF
0	Unemployment rate	20.846743
1	Inflation rate	1.865138
2	GDP	1.205098
3	Curricular units 1st sem (grade)	30.412719
4	Curricular units 2nd sem (grade)	28.566122
5	Curricular units 1st sem (credited)	18.421089
6	Curricular units 1st sem (enrolled)	177.375380
7	Curricular units 1st sem (evaluations)	19.321575
8	Curricular units 1st sem (without evaluations)	1.811622
9	Curricular units 1st sem (approved)	51.125862
10	Curricular units 2nd sem (credited)	14.311660
11	Curricular units 2nd sem (enrolled)	151.003005
12	Curricular units 2nd sem (evaluations)	17.230826
13	Curricular units 2nd sem (approved)	38.994255
14	Curricular units 2nd sem (without evaluations)	1.694993
15	Nationality	2.967007
16	Mother's qualification	3.869211
17	Father's qualification	4.552378
18	Mother's occupation	5.628926
19	Father's occupation	5.720413
20	Admission grade	29.621152
21	Application mode	2.797471
22	Application order	2.991427
23	Course	41.521306
24	International	2.830577

[348]:
```
# removing the features that has very high VIF score
features_to_remove = ['Curricular units 2nd sem (credited)',
                      'Curricular units 2nd sem (enrolled)','Curricular units␣
 ↪2nd sem (evaluations)',
                      'Curricular units 2nd sem (approved)','Curricular units␣
 ↪2nd sem (without evaluations)',
                       "International",'Admission grade',"Mother's occupation",
                      "Mother's qualification",'Curricular units 1st sem␣
 ↪(approved)',
                     'Curricular units 1st sem (enrolled)']
```

[351]:
```
# Dropping the features that are highly Correlated
df = df.drop(features_to_remove, axis=1)
```

[352]:
```
#checking VIF after dropping highly correlated features
df_columns = df[['Inflation rate', 'GDP', "Curricular units 1st sem␣
 ↪(evaluations)",
                 "Curricular units 1st sem (credited)",
                 'Curricular units 1st sem (grade)','Curricular units 2nd sem␣
 ↪(grade)',
```

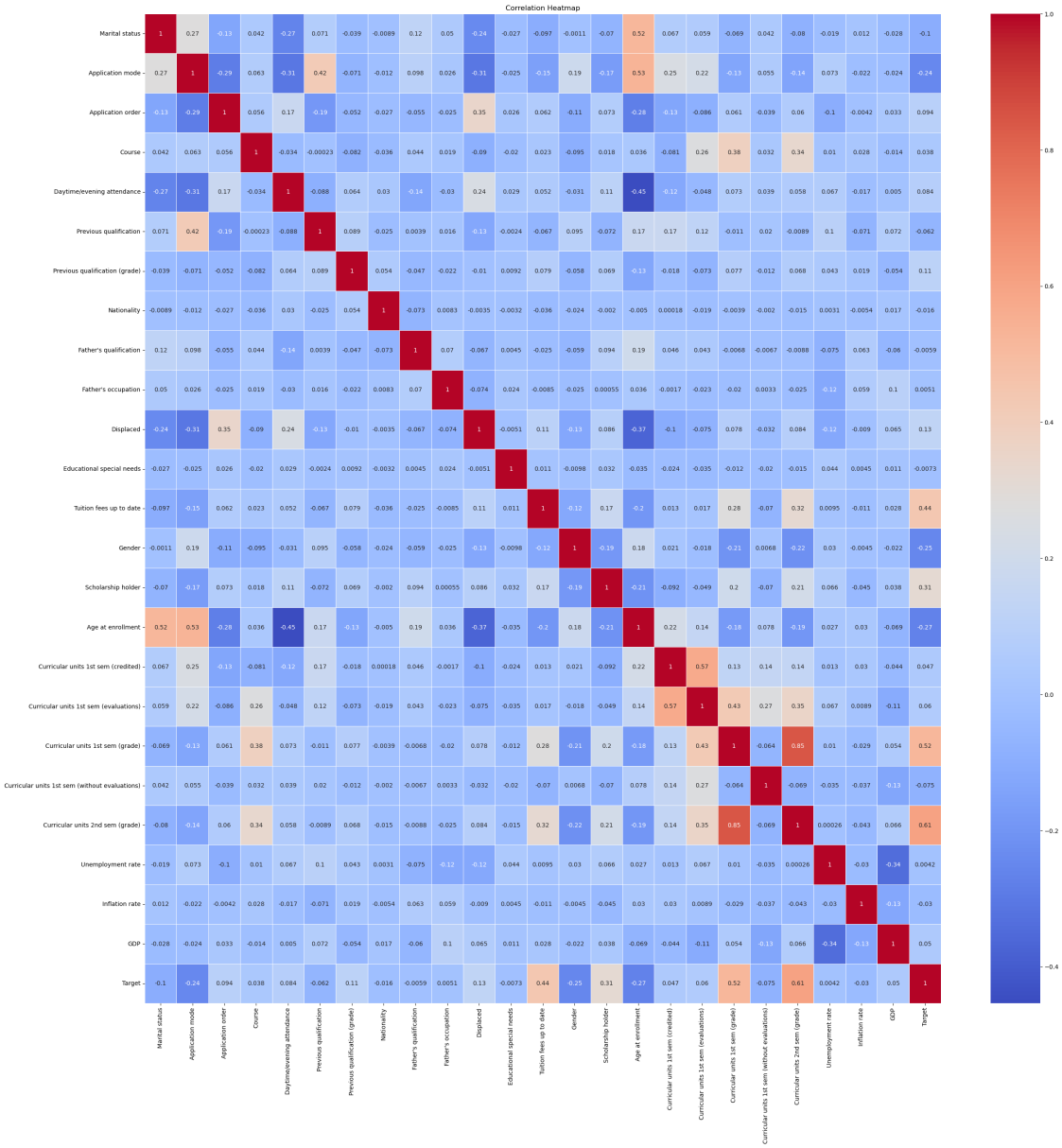
```
                'Curricular units 1st sem (without evaluations)',
                'Nationality', "Father's qualification",
                "Father's occupation",
                'Application mode', 'Application order',␣
 ↪'Course','Unemployment rate']].copy()
```

[353]:
```
X = df.iloc[:,:-1]
calc_vif(df_columns)
```

[353]:

	variables	VIF
0	Inflation rate	1.818291
1	GDP	1.129987
2	Curricular units 1st sem (evaluations)	10.038430
3	Curricular units 1st sem (credited)	1.846973
4	Curricular units 1st sem (grade)	21.426460
5	Curricular units 2nd sem (grade)	15.573070
6	Curricular units 1st sem (without evaluations)	1.190194
7	Nationality	1.077559
8	Father's qualification	3.112137
9	Father's occupation	1.237750
10	Application mode	2.666910
11	Application order	2.812605
12	Course	19.847677
13	Unemployment rate	11.947685

[354]:
```
#correlation heatmap
plt.figure(figsize=(30, 30))
sns.heatmap(df.corr() , annot=True, cmap='coolwarm', linewidths=0.5)
plt.title('Correlation Heatmap')
plt.show()
```



[355]:
```
print("\nCounts of targets in dataset:")
print("Target 0:", sum(df['Target'] == 0))
print("Target 1:", sum(df['Target'] == 1))
```

```
Counts of targets in dataset:
Target 0: 1421
Target 1: 2209
```

[356]:
```
# Calculate counts
counts = df['Target'].value_counts()
print("Target:\n", counts)

# Calculate percentages
percentages = df['Target'].value_counts(normalize=True) * 100
print("\nTargetPercentages:\n", percentages)
```

```
Target:
 1    2209
0     1421
Name: Target, dtype: int64

TargetPercentages:
 1    60.853994
0    39.146006
Name: Target, dtype: float64
```

1.5 Model Selection and Training

1.5.1 Apply SMOTETomek for Resampling

Use SMOTETomek to handle class imbalance by oversampling the minority class and undersampling the majority class.

[390]:
```
# test and train sets
X = df.drop('Target', axis=1)
y = df['Target']

smk = SMOTETomek(random_state=42)
X_res, y_res = smk.fit_resample(X, y)

X_train, X_test, y_train, y_test = train_test_split(X_res, y_res, test_size=0.
 ↪20, random_state=42)
print("X_train : ",X_train.shape)
print("X_test : ",X_test.shape)
print("y_train : ",y_train.shape)
print("y_test : ",y_test.shape)
```

```
X_train :  (3323, 25)
X_test :  (831, 25)
y_train :  (3323,)
y_test :  (831,)
```

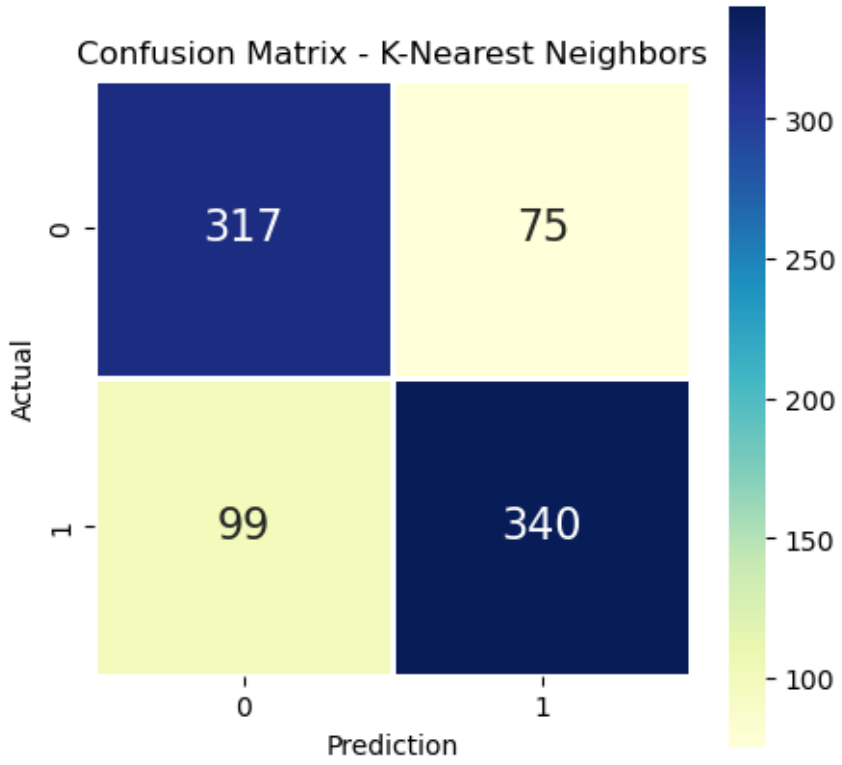
[391]:
```
algorithms = ['K-Nearest Neighbors', 'Logistic Regression', 'Decision Tree',
              'Random Forest','Easy Ensemble Classifier','AdaBoost␣
 ↪Classifier','SVM']
accuracies = []
precisions = []
```

```
recalls = []
f1_scores = []

#storing the mertrics of all algorithms
def append_metrics(accuracy, precision, recall, f1):
    accuracies.append(accuracy)
    precisions.append(precision)
    recalls.append(recall)
    f1_scores.append(f1)
```

1.5.2 K-Nearest Neighbors

[392]:
```
knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)

y_pred1 = knn.predict(X_test)


accuracy = accuracy_score(y_test, y_pred1)
precision, recall, f1, _ = precision_recall_fscore_support(y_test, y_pred1,␣
 ↪average='weighted')

print(f"Accuracy: {accuracy:.2f}")
print(f"Precision: {precision:.2f}")
print(f"Recall: {recall:.2f}")
print(f"F1-score: {f1:.2f}")

append_metrics(accuracy, precision, recall, f1)

cm = confusion_matrix(y_test, y_pred1)

fig, ax = plt.subplots(figsize=(5, 5))
sns.heatmap(cm, fmt=".0f", cmap="YlGnBu", linewidth=1, square=True, annot=True,␣
 ↪annot_kws={"fontsize": 16}, ax=ax)
ax.set_xlabel("Prediction")
ax.set_ylabel("Actual")
plt.title("Confusion Matrix - K-Nearest Neighbors")

plt.show()
```

```
Accuracy: 0.79
Precision: 0.79
Recall: 0.79
F1-score: 0.79
```

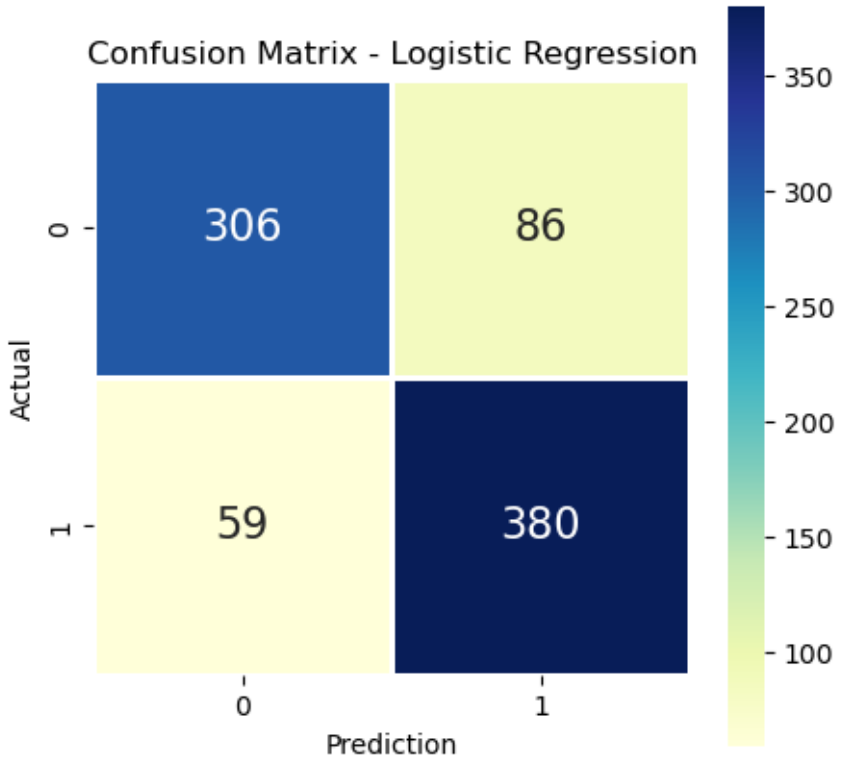


1.5.3 Logistic Regression

[393]:
```
logistic_model = LogisticRegression(C=1, random_state=42)
logistic_model.fit(X_train, y_train)

y_pred2 = logistic_model.predict(X_test)

accuracy = accuracy_score(y_test, y_pred2)
precision, recall, f1, _ = precision_recall_fscore_support(y_test, y_pred2,␣
↪average='weighted')

print("\nLogistic Regression with Hyperparameter Tuning:")
print(f"Accuracy: {accuracy:.2f}")
print(f"Precision: {precision:.2f}")
print(f"Recall: {recall:.2f}")
print(f"F1-score: {f1:.2f}")

append_metrics(accuracy, precision, recall, f1)

cm_logistic_tuned = confusion_matrix(y_test, y_pred2)
fig, ax = plt.subplots(figsize=(5, 5))
```

```
sns.heatmap(cm_logistic_tuned, fmt=".0f", cmap="YlGnBu", linewidth=1,␣
 ↪square=True, annot=True, annot_kws={"fontsize": 16}, ax=ax)
ax.set_xlabel("Prediction")
ax.set_ylabel("Actual")
plt.title("Confusion Matrix - Logistic Regression")
plt.show()
```

```
Logistic Regression with Hyperparameter Tuning:
Accuracy: 0.83
Precision: 0.83
Recall: 0.83
F1-score: 0.83
```

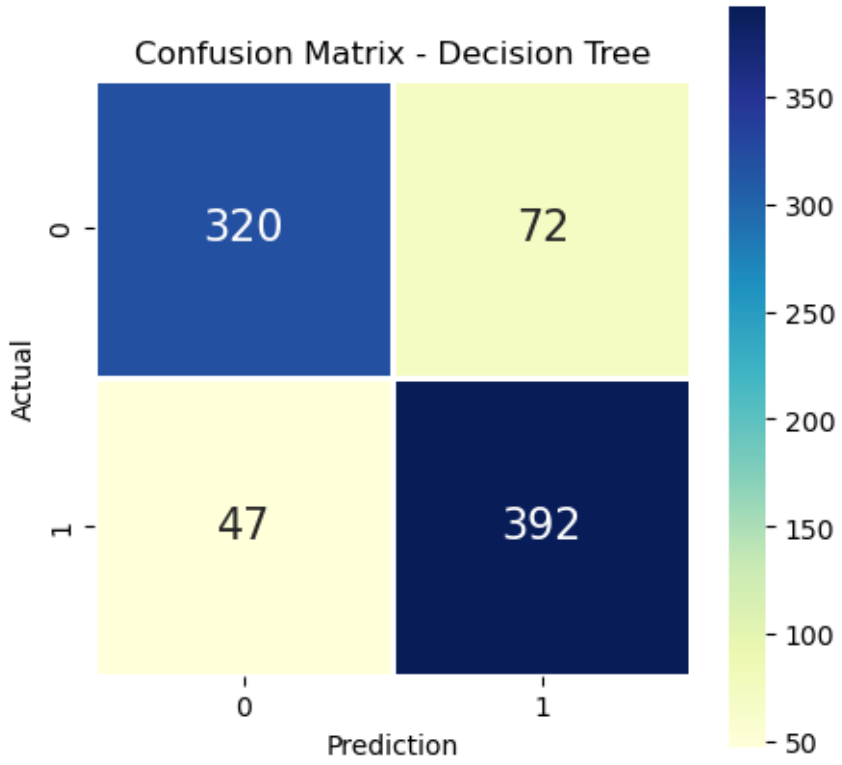


1.5.4 Decision Tree

[394]:
```
tree_clf = DecisionTreeClassifier(max_features='auto',
                                  ccp_alpha=0.001,
                                  max_depth=15,
                                  criterion='gini',random_state=1234 )
tree_clf.fit(X_train, y_train)
```

```
y_pred3 = tree_clf.predict(X_test)

accuracy = accuracy_score(y_test, y_pred3)
precision, recall, f1, _ = precision_recall_fscore_support(y_test, y_pred3,␣
 ↪average='weighted')

print(f"Accuracy: {accuracy:.2f}")
print(f"Precision: {precision:.2f}")
print(f"Recall: {recall:.2f}")
print(f"F1-score: {f1:.2f}")

append_metrics(accuracy, precision, recall, f1)

cm = confusion_matrix(y_test, y_pred3)

fig, ax = plt.subplots(figsize=(5, 5))
sns.heatmap(cm, fmt=".0f", cmap="YlGnBu", linewidth=1, square=True, annot=True,␣
 ↪annot_kws={"fontsize": 16}, ax=ax)

ax.set_xlabel("Prediction")
ax.set_ylabel("Actual")
plt.title("Confusion Matrix - Decision Tree")
plt.show()
```

```
Accuracy: 0.86
Precision: 0.86
Recall: 0.86
F1-score: 0.86
```

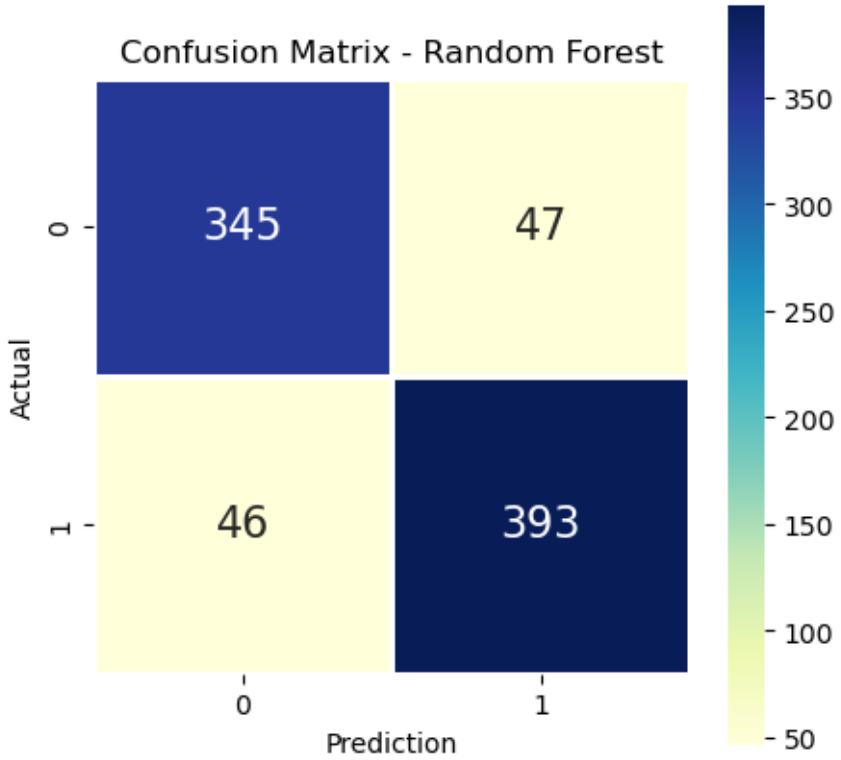


1.5.5 Random Forest

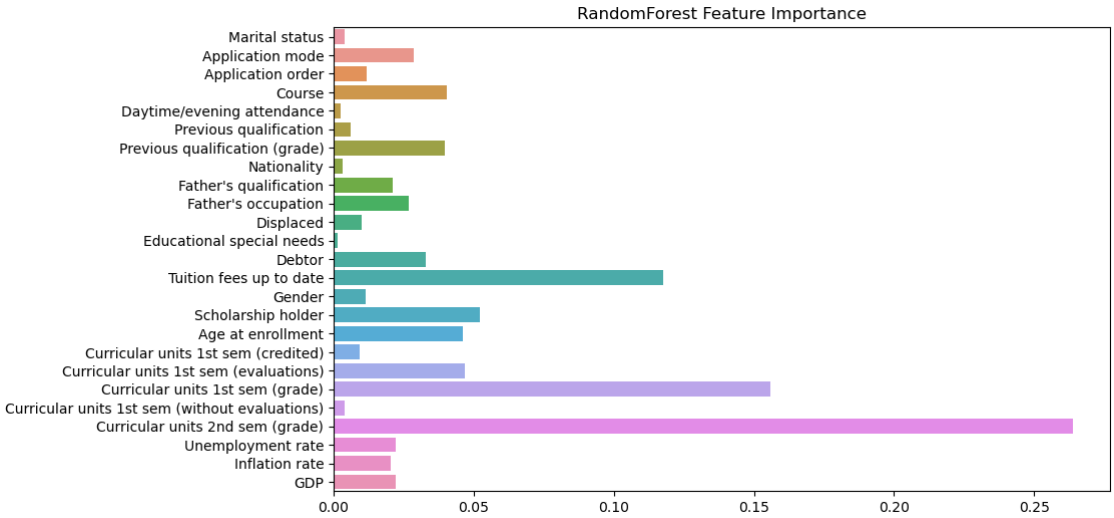
[395]:
```
rf = RandomForestClassifier()
param_grid = {'n_estimators': [50, 100, 150, 200], 'max_depth': [5, 10, 20,␣
 ↪30],'max_features':['sqrt', 'log2']}

grid_search = GridSearchCV(rf, param_grid, cv=5, scoring='accuracy')
grid_search.fit(X_train, y_train)
best_params  = grid_search.best_params_
print("Best parameters:", best_params)


best_rf = RandomForestClassifier(n_estimators=best_params['n_estimators'],␣
 ↪max_depth=best_params['max_depth'],
                                 max_features=best_params['max_features'])
best_rf.fit(X_train, y_train)

y_pred4 = best_rf.predict(X_test)

accuracy = accuracy_score(y_test, y_pred4)
precision, recall, f1, _ = precision_recall_fscore_support(y_test, y_pred4,␣
 ↪average='weighted')
```

```
print(f"Accuracy: {accuracy:.2f}")
print(f"Precision: {precision:.2f}")
print(f"Recall: {recall:.2f}")
print(f"F1-score: {f1:.2f}")

append_metrics(accuracy, precision, recall, f1)

cm = confusion_matrix(y_test, y_pred4)

fig, ax = plt.subplots(figsize=(5, 5))
sns.heatmap(cm, fmt=".0f", cmap="YlGnBu", linewidth=1, square=True, annot=True,␣
 ↪annot_kws={"fontsize": 16}, ax=ax)

ax.set_xlabel("Prediction")
ax.set_ylabel("Actual")
plt.title("Confusion Matrix - Random Forest")
plt.show()
```

```
Best parameters: {'max_depth': 30, 'max_features': 'sqrt', 'n_estimators': 150}
Accuracy: 0.89
Precision: 0.89
Recall: 0.89
F1-score: 0.89
```



[396]:
```
#Feature importance
feature_importances = best_rf.feature_importances_
plt.figure(figsize=(10, 6))
sns.barplot(x=feature_importances, y=X.columns)
plt.title("RandomForest Feature Importance")
plt.show()
```

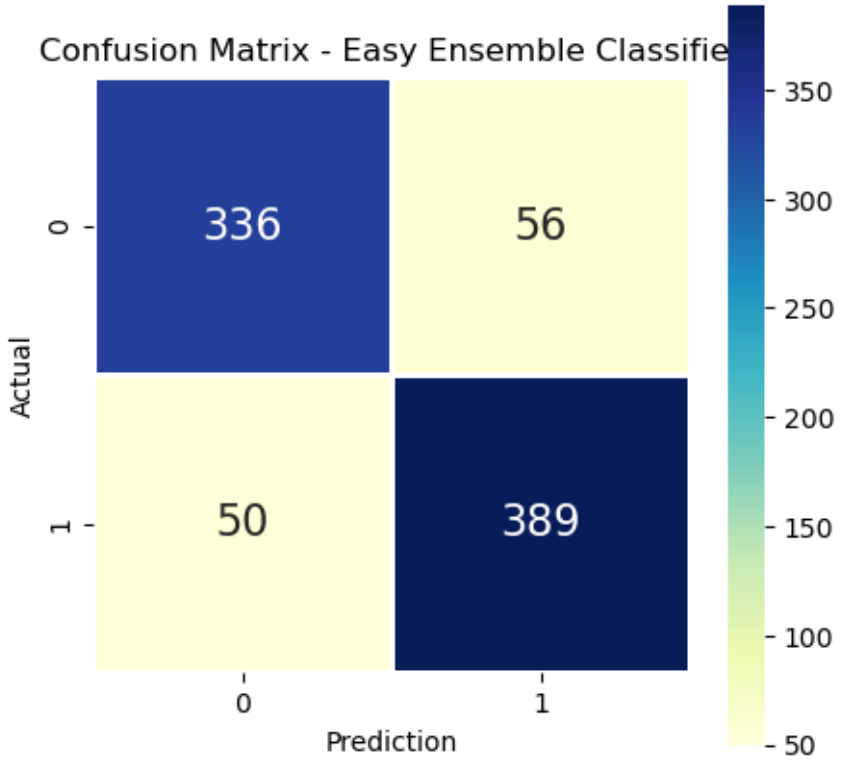


1.5.6 Easy Ensemble Classifier

[397]:
```
eec = EasyEnsembleClassifier(random_state=42)
eec.fit(X_train, y_train)
y_pred5 = eec.predict(X_test)

accuracy = accuracy_score(y_test, y_pred5)
precision, recall, f1, _ = precision_recall_fscore_support(y_test, y_pred5,␣
 ↪average='weighted')

print(f"Accuracy: {accuracy:.2f}")
print(f"Precision: {precision:.2f}")
print(f"Recall: {recall:.2f}")
print(f"F1-score: {f1:.2f}")

append_metrics(accuracy, precision, recall, f1)

cm = confusion_matrix(y_test, y_pred5)
```

```
fig, ax = plt.subplots(figsize=(5, 5))
sns.heatmap(cm, fmt=".0f", cmap="YlGnBu", linewidth=1, square=True, annot=True,␣
 ↪annot_kws={"fontsize": 16}, ax=ax)

ax.set_xlabel("Prediction")
ax.set_ylabel("Actual")
plt.title("Confusion Matrix - Easy Ensemble Classifier")
plt.show()
```

```
Accuracy: 0.87
Precision: 0.87
Recall: 0.87
F1-score: 0.87
```

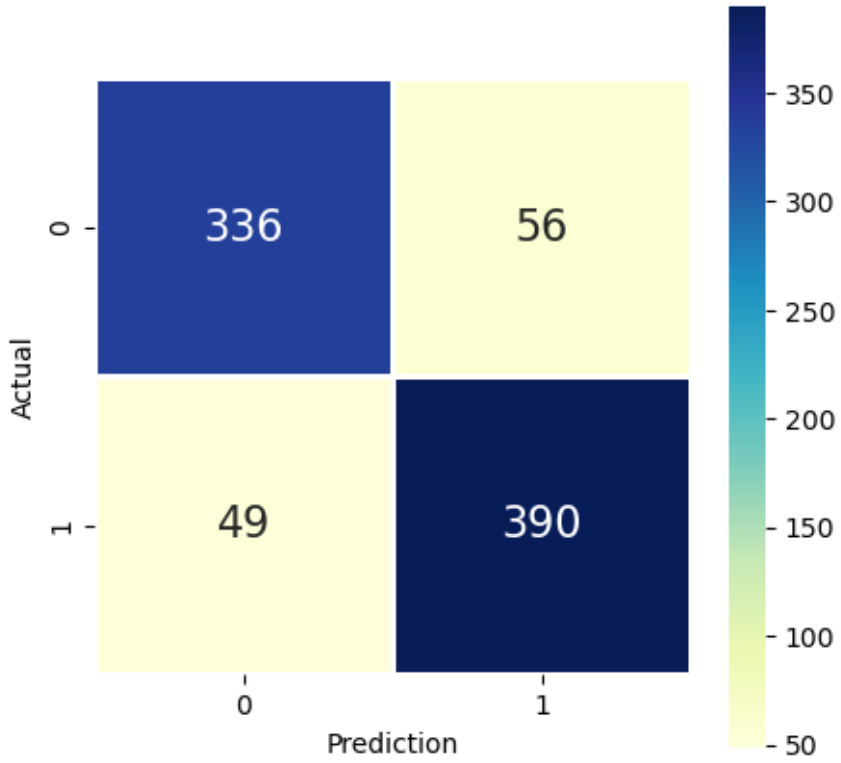


1.5.7 AdaBoost Classifier

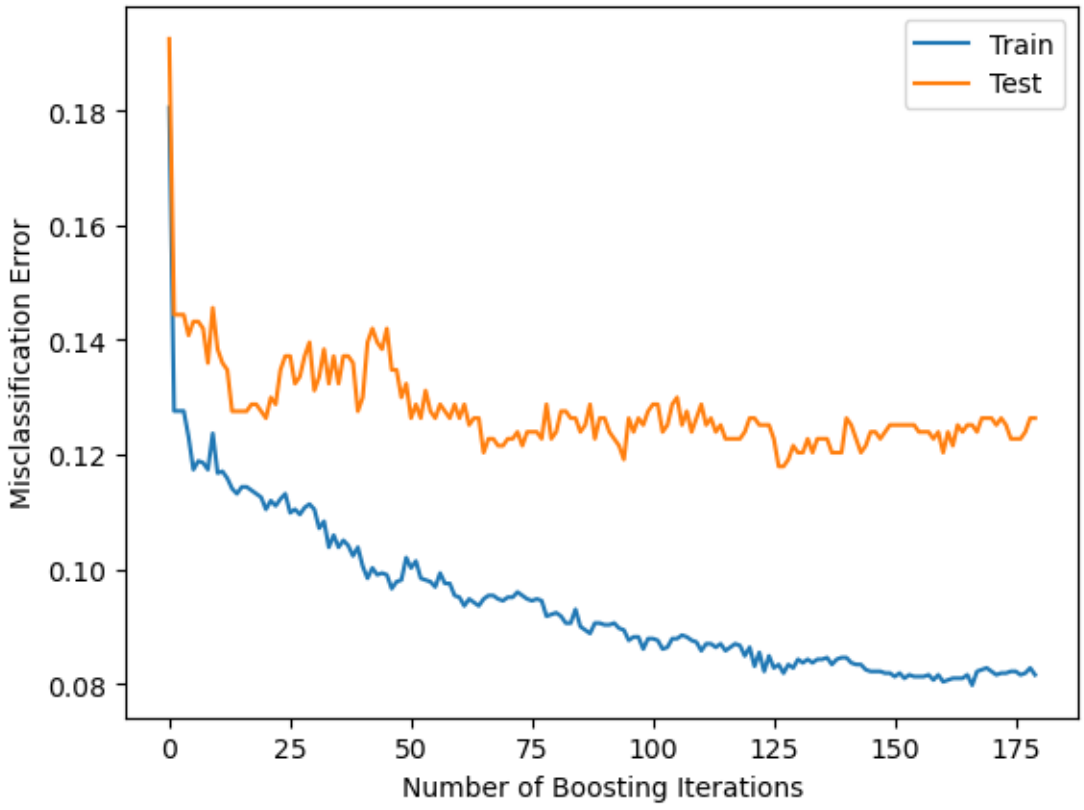
[398]:
```
ada=AdaBoostClassifier()

param_grid={'n_estimators' : [50, 70, 90, 120, 180, 200],
                          'learning_rate' : [0.001, 0.01, 0.1, 1, 10]}
grid_search = GridSearchCV(ada,param_grid, n_jobs = -1, cv = 5, verbose = 1)
grid_search.fit(X_train, y_train)
```

```
best_params  = grid_search.best_params_
print("Best parameters:", best_params)



AdaBoost = AdaBoostClassifier(n_estimators=best_params['n_estimators'],␣
 ↪learning_rate=best_params['learning_rate'],
                               random_state=0)
AdaBoost.fit(X_train, y_train)

y_pred6 = AdaBoost.predict(X_test)

accuracy = accuracy_score(y_test, y_pred6)
precision, recall, f1, _ = precision_recall_fscore_support(y_test, y_pred6,␣
 ↪average='weighted')

print(f"Accuracy: {accuracy:.2f}")
print(f"Precision: {precision:.2f}")
print(f"Recall: {recall:.2f}")
print(f"F1-score: {f1:.2f}")

append_metrics(accuracy, precision, recall, f1)

cm = confusion_matrix(y_test, y_pred6)

fig, ax = plt.subplots(figsize=(5, 5))
sns.heatmap(cm, fmt=".0f", cmap="YlGnBu", linewidth=1, square=True, annot=True,␣
 ↪annot_kws={"fontsize": 16}, ax=ax)

ax.set_xlabel("Prediction")
ax.set_ylabel("Actual")
plt.show()
```

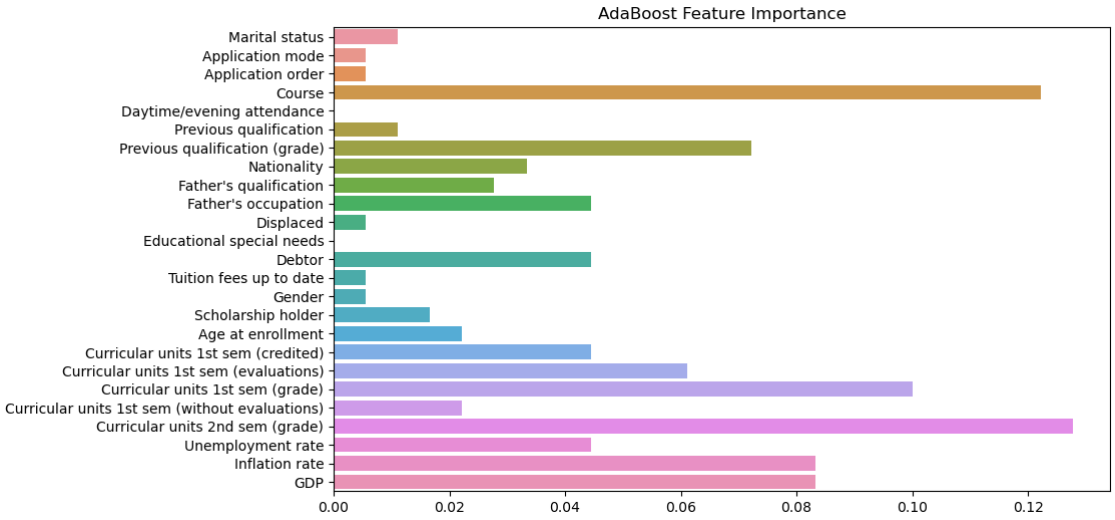
```
Fitting 5 folds for each of 30 candidates, totalling 150 fits
Best parameters: {'learning_rate': 1, 'n_estimators': 180}
Accuracy: 0.87
Precision: 0.87
Recall: 0.87
F1-score: 0.87
```



[399]:
```
# misclassification error for train data
staged_score = AdaBoost.staged_score(X_train,y_train)
misclassification_error = []

for i, score in enumerate(staged_score):
    misclassification_error.append(1-score)

# misclassification error for test data
staged_score_test = AdaBoost.staged_score(X_test,y_test)
misclassification_error_test = []
for i, score in enumerate(staged_score_test):
     misclassification_error_test.append(1-score)

plt.plot(misclassification_error,label="Train")
plt.plot(misclassification_error_test, label="Test")
plt.xlabel('Number of Boosting Iterations')
plt.ylabel('Misclassification Error')
plt.legend()
plt.show()
```



[400]:
```
# Get feature importances
importances = AdaBoost.feature_importances_

plt.figure(figsize=(10, 6))
sns.barplot(x=importances, y=X.columns)
plt.title("AdaBoost Feature Importance")
plt.show()
```



1.5.8 Support Vector Machine

[401]:
```
svc = SVC()
param_grid = {'C': [0.1, 1, 10, 100, 1000],
              'gamma': [1, 0.1, 0.01, 0.001, 0.0001],
              'kernel': ['rbf']}
grid_search = GridSearchCV(svc, param_grid, cv=5, scoring='accuracy',␣
 ↪refit=True, verbose=3)
grid_search.fit(X_train, y_train)

best_params_SVC = grid_search.best_params_
print("Best Parameters:", best_params_SVC)
```

```
Fitting 5 folds for each of 25 candidates, totalling 125 fits
[CV 1/5] END …C=0.1, gamma=1, kernel=rbf;, score=0.507 total time=   0.2s
[CV 2/5] END …C=0.1, gamma=1, kernel=rbf;, score=0.507 total time=   0.2s
[CV 3/5] END …C=0.1, gamma=1, kernel=rbf;, score=0.507 total time=   0.2s
[CV 4/5] END …C=0.1, gamma=1, kernel=rbf;, score=0.508 total time=   0.2s
[CV 5/5] END …C=0.1, gamma=1, kernel=rbf;, score=0.508 total time=   0.2s
[CV 1/5] END …C=0.1, gamma=0.1, kernel=rbf;, score=0.507 total time=   0.2s
[CV 2/5] END …C=0.1, gamma=0.1, kernel=rbf;, score=0.507 total time=   0.2s
[CV 3/5] END …C=0.1, gamma=0.1, kernel=rbf;, score=0.507 total time=   0.3s
[CV 4/5] END …C=0.1, gamma=0.1, kernel=rbf;, score=0.508 total time=   0.2s
[CV 5/5] END …C=0.1, gamma=0.1, kernel=rbf;, score=0.508 total time=   0.2s
[CV 1/5] END …C=0.1, gamma=0.01, kernel=rbf;, score=0.620 total time=   0.3s
[CV 2/5] END …C=0.1, gamma=0.01, kernel=rbf;, score=0.606 total time=   0.3s
[CV 3/5] END …C=0.1, gamma=0.01, kernel=rbf;, score=0.588 total time=   0.3s
[CV 4/5] END …C=0.1, gamma=0.01, kernel=rbf;, score=0.584 total time=   0.3s
[CV 5/5] END …C=0.1, gamma=0.01, kernel=rbf;, score=0.596 total time=   0.3s
[CV 1/5] END …C=0.1, gamma=0.001, kernel=rbf;, score=0.732 total time=   0.2s
```

```
[CV 2/5] END …C=0.1, gamma=0.001, kernel=rbf;, score=0.752 total time=   0.2s
[CV 3/5] END …C=0.1, gamma=0.001, kernel=rbf;, score=0.708 total time=   0.2s
[CV 4/5] END …C=0.1, gamma=0.001, kernel=rbf;, score=0.715 total time=   0.2s
[CV 5/5] END …C=0.1, gamma=0.001, kernel=rbf;, score=0.761 total time=   0.2s
[CV 1/5] END …C=0.1, gamma=0.0001, kernel=rbf;, score=0.657 total time=   0.2s
[CV 2/5] END …C=0.1, gamma=0.0001, kernel=rbf;, score=0.647 total time=   0.2s
[CV 3/5] END …C=0.1, gamma=0.0001, kernel=rbf;, score=0.678 total time=   0.2s
[CV 4/5] END …C=0.1, gamma=0.0001, kernel=rbf;, score=0.675 total time=   0.2s
[CV 5/5] END …C=0.1, gamma=0.0001, kernel=rbf;, score=0.681 total time=   0.2s
[CV 1/5] END …C=1, gamma=1, kernel=rbf;, score=0.514 total time=   0.2s
[CV 2/5] END …C=1, gamma=1, kernel=rbf;, score=0.514 total time=   0.2s
[CV 3/5] END …C=1, gamma=1, kernel=rbf;, score=0.510 total time=   0.2s
[CV 4/5] END …C=1, gamma=1, kernel=rbf;, score=0.508 total time=   0.2s
[CV 5/5] END …C=1, gamma=1, kernel=rbf;, score=0.514 total time=   0.2s
[CV 1/5] END …C=1, gamma=0.1, kernel=rbf;, score=0.708 total time=   0.3s
[CV 2/5] END …C=1, gamma=0.1, kernel=rbf;, score=0.711 total time=   0.3s
[CV 3/5] END …C=1, gamma=0.1, kernel=rbf;, score=0.738 total time=   0.3s
[CV 4/5] END …C=1, gamma=0.1, kernel=rbf;, score=0.706 total time=   0.3s
[CV 5/5] END …C=1, gamma=0.1, kernel=rbf;, score=0.705 total time=   0.3s
[CV 1/5] END …C=1, gamma=0.01, kernel=rbf;, score=0.798 total time=   0.3s
[CV 2/5] END …C=1, gamma=0.01, kernel=rbf;, score=0.800 total time=   0.3s
[CV 3/5] END …C=1, gamma=0.01, kernel=rbf;, score=0.791 total time=   0.3s
[CV 4/5] END …C=1, gamma=0.01, kernel=rbf;, score=0.803 total time=   0.3s
[CV 5/5] END …C=1, gamma=0.01, kernel=rbf;, score=0.801 total time=   0.3s
[CV 1/5] END …C=1, gamma=0.001, kernel=rbf;, score=0.830 total time=   0.2s
[CV 2/5] END …C=1, gamma=0.001, kernel=rbf;, score=0.827 total time=   0.2s
[CV 3/5] END …C=1, gamma=0.001, kernel=rbf;, score=0.797 total time=   0.2s
[CV 4/5] END …C=1, gamma=0.001, kernel=rbf;, score=0.825 total time=   0.2s
[CV 5/5] END …C=1, gamma=0.001, kernel=rbf;, score=0.824 total time=   0.2s
[CV 1/5] END …C=1, gamma=0.0001, kernel=rbf;, score=0.806 total time=   0.2s
[CV 2/5] END …C=1, gamma=0.0001, kernel=rbf;, score=0.802 total time=   0.2s
[CV 3/5] END …C=1, gamma=0.0001, kernel=rbf;, score=0.792 total time=   0.1s
[CV 4/5] END …C=1, gamma=0.0001, kernel=rbf;, score=0.816 total time=   0.1s
[CV 5/5] END …C=1, gamma=0.0001, kernel=rbf;, score=0.821 total time=   0.2s
[CV 1/5] END …C=10, gamma=1, kernel=rbf;, score=0.516 total time=   0.3s
[CV 2/5] END …C=10, gamma=1, kernel=rbf;, score=0.517 total time=   0.2s
[CV 3/5] END …C=10, gamma=1, kernel=rbf;, score=0.513 total time=   0.2s
[CV 4/5] END …C=10, gamma=1, kernel=rbf;, score=0.508 total time=   0.2s
[CV 5/5] END …C=10, gamma=1, kernel=rbf;, score=0.515 total time=   0.2s
[CV 1/5] END …C=10, gamma=0.1, kernel=rbf;, score=0.717 total time=   0.3s
[CV 2/5] END …C=10, gamma=0.1, kernel=rbf;, score=0.713 total time=   0.3s
[CV 3/5] END …C=10, gamma=0.1, kernel=rbf;, score=0.737 total time=   0.3s
[CV 4/5] END …C=10, gamma=0.1, kernel=rbf;, score=0.709 total time=   0.3s
[CV 5/5] END …C=10, gamma=0.1, kernel=rbf;, score=0.702 total time=   0.3s
[CV 1/5] END …C=10, gamma=0.01, kernel=rbf;, score=0.820 total time=   0.3s
[CV 2/5] END …C=10, gamma=0.01, kernel=rbf;, score=0.818 total time=   0.3s
[CV 3/5] END …C=10, gamma=0.01, kernel=rbf;, score=0.829 total time=   0.3s
[CV 4/5] END …C=10, gamma=0.01, kernel=rbf;, score=0.809 total time=   0.3s
```

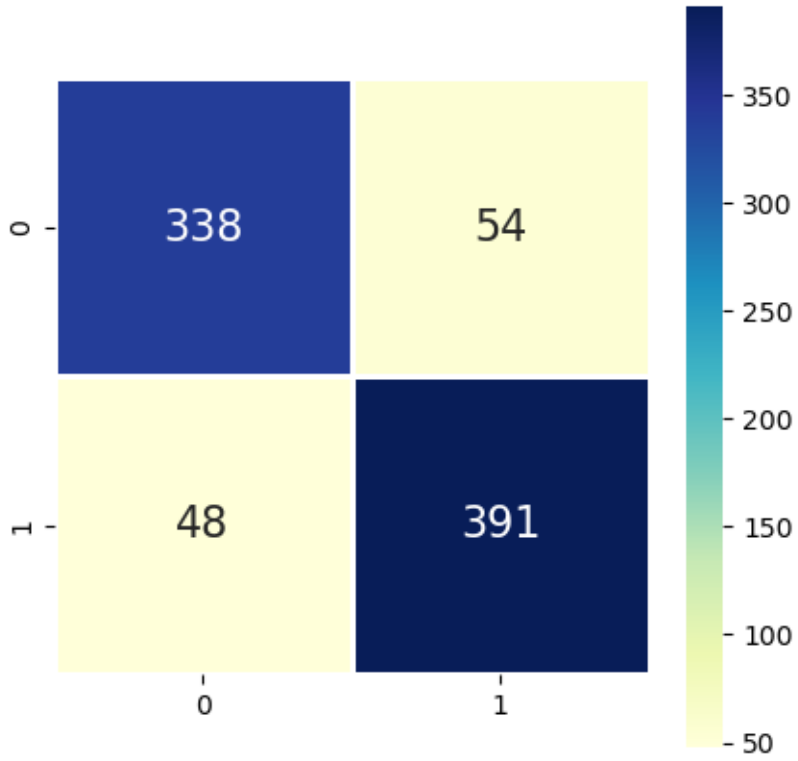
```
[CV 5/5] END …C=10, gamma=0.01, kernel=rbf;, score=0.824 total time=   0.3s
[CV 1/5] END …C=10, gamma=0.001, kernel=rbf;, score=0.863 total time=   0.2s
[CV 2/5] END …C=10, gamma=0.001, kernel=rbf;, score=0.856 total time=   0.1s
[CV 3/5] END …C=10, gamma=0.001, kernel=rbf;, score=0.841 total time=   0.1s
[CV 4/5] END …C=10, gamma=0.001, kernel=rbf;, score=0.872 total time=   0.1s
[CV 5/5] END …C=10, gamma=0.001, kernel=rbf;, score=0.855 total time=   0.1s
[CV 1/5] END …C=10, gamma=0.0001, kernel=rbf;, score=0.833 total time=   0.1s
[CV 2/5] END …C=10, gamma=0.0001, kernel=rbf;, score=0.821 total time=   0.1s
[CV 3/5] END …C=10, gamma=0.0001, kernel=rbf;, score=0.830 total time=   0.1s
[CV 4/5] END …C=10, gamma=0.0001, kernel=rbf;, score=0.845 total time=   0.1s
[CV 5/5] END …C=10, gamma=0.0001, kernel=rbf;, score=0.840 total time=   0.1s
[CV 1/5] END …C=100, gamma=1, kernel=rbf;, score=0.516 total time=   0.2s
[CV 2/5] END …C=100, gamma=1, kernel=rbf;, score=0.517 total time=   0.2s
[CV 3/5] END …C=100, gamma=1, kernel=rbf;, score=0.513 total time=   0.2s
[CV 4/5] END …C=100, gamma=1, kernel=rbf;, score=0.508 total time=   0.2s
[CV 5/5] END …C=100, gamma=1, kernel=rbf;, score=0.515 total time=   0.2s
[CV 1/5] END …C=100, gamma=0.1, kernel=rbf;, score=0.717 total time=   0.3s
[CV 2/5] END …C=100, gamma=0.1, kernel=rbf;, score=0.713 total time=   0.3s
[CV 3/5] END …C=100, gamma=0.1, kernel=rbf;, score=0.737 total time=   0.3s
[CV 4/5] END …C=100, gamma=0.1, kernel=rbf;, score=0.709 total time=   0.3s
[CV 5/5] END …C=100, gamma=0.1, kernel=rbf;, score=0.702 total time=   0.3s
[CV 1/5] END …C=100, gamma=0.01, kernel=rbf;, score=0.820 total time=   0.3s
[CV 2/5] END …C=100, gamma=0.01, kernel=rbf;, score=0.818 total time=   0.3s
[CV 3/5] END …C=100, gamma=0.01, kernel=rbf;, score=0.823 total time=   0.3s
[CV 4/5] END …C=100, gamma=0.01, kernel=rbf;, score=0.809 total time=   0.3s
[CV 5/5] END …C=100, gamma=0.01, kernel=rbf;, score=0.828 total time=   0.5s
[CV 1/5] END …C=100, gamma=0.001, kernel=rbf;, score=0.869 total time=   0.2s
[CV 2/5] END …C=100, gamma=0.001, kernel=rbf;, score=0.839 total time=   0.2s
[CV 3/5] END …C=100, gamma=0.001, kernel=rbf;, score=0.835 total time=   0.2s
[CV 4/5] END …C=100, gamma=0.001, kernel=rbf;, score=0.861 total time=   0.2s
[CV 5/5] END …C=100, gamma=0.001, kernel=rbf;, score=0.846 total time=   0.2s
[CV 1/5] END …C=100, gamma=0.0001, kernel=rbf;, score=0.875 total time=   0.1s
[CV 2/5] END …C=100, gamma=0.0001, kernel=rbf;, score=0.871 total time=   0.1s
[CV 3/5] END …C=100, gamma=0.0001, kernel=rbf;, score=0.872 total time=   0.1s
[CV 4/5] END …C=100, gamma=0.0001, kernel=rbf;, score=0.881 total time=   0.1s
[CV 5/5] END …C=100, gamma=0.0001, kernel=rbf;, score=0.869 total time=   0.1s
[CV 1/5] END …C=1000, gamma=1, kernel=rbf;, score=0.516 total time=   0.2s
[CV 2/5] END …C=1000, gamma=1, kernel=rbf;, score=0.517 total time=   0.2s
[CV 3/5] END …C=1000, gamma=1, kernel=rbf;, score=0.513 total time=   0.2s
[CV 4/5] END …C=1000, gamma=1, kernel=rbf;, score=0.508 total time=   0.2s
[CV 5/5] END …C=1000, gamma=1, kernel=rbf;, score=0.515 total time=   0.2s
[CV 1/5] END …C=1000, gamma=0.1, kernel=rbf;, score=0.717 total time=   0.3s
[CV 2/5] END …C=1000, gamma=0.1, kernel=rbf;, score=0.713 total time=   0.3s
[CV 3/5] END …C=1000, gamma=0.1, kernel=rbf;, score=0.737 total time=   0.3s
[CV 4/5] END …C=1000, gamma=0.1, kernel=rbf;, score=0.709 total time=   0.3s
[CV 5/5] END …C=1000, gamma=0.1, kernel=rbf;, score=0.702 total time=   0.3s
[CV 1/5] END …C=1000, gamma=0.01, kernel=rbf;, score=0.820 total time=   0.3s
[CV 2/5] END …C=1000, gamma=0.01, kernel=rbf;, score=0.818 total time=   0.3s
```

```
[CV 3/5] END …C=1000, gamma=0.01, kernel=rbf;, score=0.823 total time=   0.3s
[CV 4/5] END …C=1000, gamma=0.01, kernel=rbf;, score=0.809 total time=   0.3s
[CV 5/5] END …C=1000, gamma=0.01, kernel=rbf;, score=0.828 total time=   0.3s
[CV 1/5] END …C=1000, gamma=0.001, kernel=rbf;, score=0.850 total time=   0.2s
[CV 2/5] END …C=1000, gamma=0.001, kernel=rbf;, score=0.821 total time=   0.2s
[CV 3/5] END …C=1000, gamma=0.001, kernel=rbf;, score=0.845 total time=   0.2s
[CV 4/5] END …C=1000, gamma=0.001, kernel=rbf;, score=0.851 total time=   0.2s
[CV 5/5] END …C=1000, gamma=0.001, kernel=rbf;, score=0.837 total time=   0.2s
[CV 1/5] END ..C=1000, gamma=0.0001, kernel=rbf;, score=0.871 total time=   0.3s
[CV 2/5] END ..C=1000, gamma=0.0001, kernel=rbf;, score=0.869 total time=   0.3s
[CV 3/5] END ..C=1000, gamma=0.0001, kernel=rbf;, score=0.865 total time=   0.3s
[CV 4/5] END ..C=1000, gamma=0.0001, kernel=rbf;, score=0.896 total time=   0.3s
[CV 5/5] END ..C=1000, gamma=0.0001, kernel=rbf;, score=0.852 total time=   0.3s
Best Parameters: {'C': 100, 'gamma': 0.0001, 'kernel': 'rbf'}
```

[402]:
```
print("Best Parameters:", best_params_SVC)

best_svc = SVC(gamma=best_params_SVC['gamma'],␣
 ↪kernel=best_params_SVC['kernel'], C=best_params_SVC['C'])
best_svc.fit(X_train, y_train)

y_pred7 = best_svc.predict(X_test)

accuracy = accuracy_score(y_test, y_pred7)
precision, recall, f1, _ = precision_recall_fscore_support(y_test, y_pred7,␣
 ↪average='weighted')

print(f"Accuracy: {accuracy:.2f}")
print(f"Precision: {precision:.2f}")
print(f"Recall: {recall:.2f}")
print(f"F1-score: {f1:.2f}")

append_metrics(accuracy, precision, recall, f1)

cm = confusion_matrix(y_test, y_pred7)

fig, ax = plt.subplots(figsize=(5, 5))
sns.heatmap(cm, fmt=".0f", cmap="YlGnBu", linewidth=1, square=True, annot=True,␣
 ↪annot_kws={"fontsize": 16}, ax=ax)
```

```
Best Parameters: {'C': 100, 'gamma': 0.0001, 'kernel': 'rbf'}
Accuracy: 0.88
Precision: 0.88
Recall: 0.88
F1-score: 0.88
```

[402]: <Axes: >



1.5.9 Result Comparison

[403]:
```
#display results
results_df = pd.DataFrame({
    'Algorithm': algorithms,
    'Accuracy': accuracies,
    'Precision': precisions,
    'Recall': recalls,
    'F1': f1
})
results_df = results_df.sort_values(by='Accuracy', ascending=False)
results_df
```

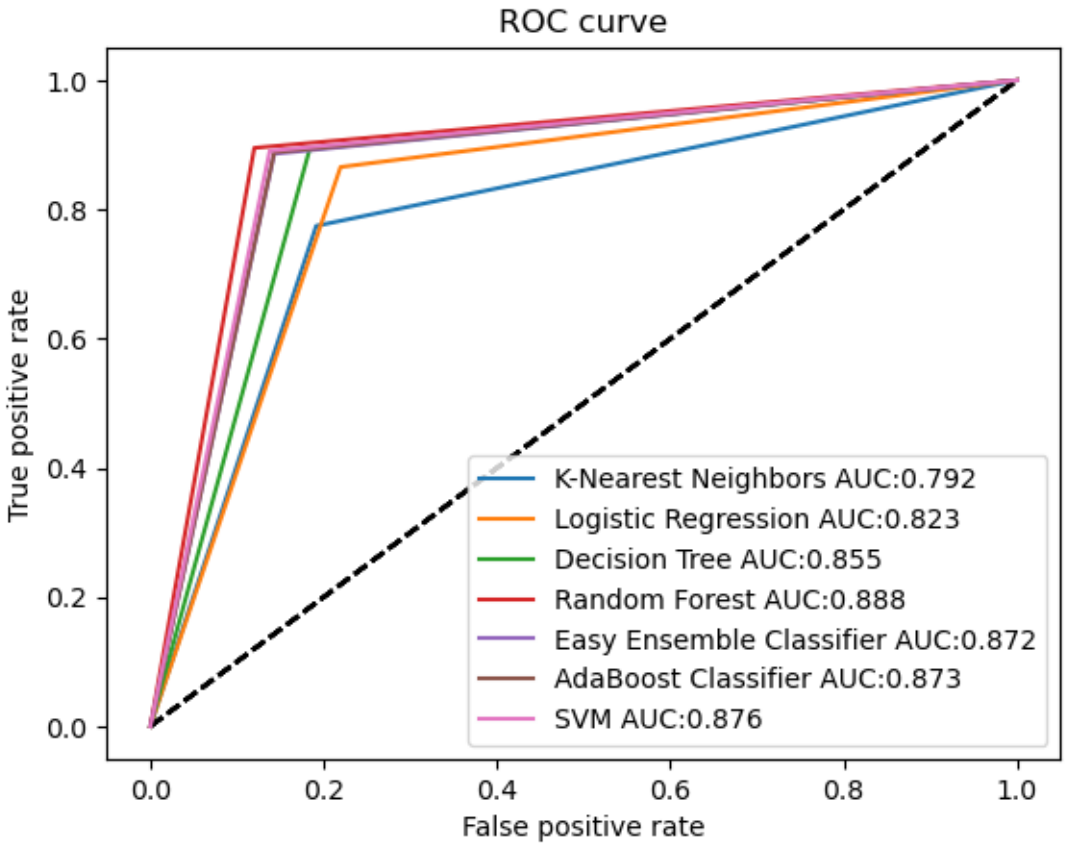
[403]:

	Algorithm	Accuracy	Precision	Recall	F1
3	Random Forest	0.888087	0.888074	0.888087	0.8772
6	SVM	0.877256	0.877235	0.877256	0.8772
5	AdaBoost Classifier	0.873646	0.873632	0.873646	0.8772
4	Easy Ensemble Classifier	0.872443	0.872416	0.872443	0.8772
2	Decision Tree	0.856799	0.857615	0.856799	0.8772
1	Logistic Regression	0.825511	0.826256	0.825511	0.8772
0	K-Nearest Neighbors	0.790614	0.792267	0.790614	0.8772

[404]:
```
#function to plot ROC curve
def show_ROCs(scores_list: list, ys_list: list, labels_list:list = None):
    """
    This function plots a couple of ROCs. Corresponding labels are optional.

    Parameters
    ----------
    scores_list : list of array-likes with scorings or predicted probabilities.
    ys_list : list of array-likes with ground true labels.
    labels_list : list of labels to be displayed in plotted graph.

    Returns
    ----------
    None

    """
    if len(scores_list) != len(ys_list):
        raise Exception('len(scores_list) != len(ys_list)')
    fpr_dict = dict()
    tpr_dict = dict()
    for x in range(len(scores_list)):
        fpr_dict[x], tpr_dict[x], _ = roc_curve(ys_list[x], scores_list[x])
    for x in range(len(scores_list)):
        try:
            plot_ROC(fpr_dict[x], tpr_dict[x], str(labels_list[x]) + ' AUC:' +␣
 ↪str(round(auc(fpr_dict[x], tpr_dict[x]),3)))
        except:
            plot_ROC(fpr_dict[x], tpr_dict[x], str(x) + ' ' +␣
 ↪str(round(auc(fpr_dict[x], tpr_dict[x]),3)))
    plt.show()

def plot_ROC(fpr, tpr, label):
    """
    This function plots a single ROC. Corresponding label is optional.
    Parameters
    ----------
    fpr : array-likes with fpr.
    tpr : array-likes with tpr.
    label : label to be displayed in plotted graph.

    Returns
    ----------
    None

    """
    plt.figure(1)
    plt.plot([0, 1], [0, 1], 'k--')
```

```
    plt.plot(fpr, tpr, label=label)
    plt.xlabel('False positive rate')
    plt.ylabel('True positive rate')
    plt.title('ROC curve')
    plt.legend(loc='best')
```

[405]:
```
show_ROCs(
    [y_pred1, y_pred2,y_pred3,y_pred4,y_pred5,y_pred6,y_pred7],
    [y_test, y_test,y_test,y_test,y_test,y_test,y_test],
    ['K-Nearest Neighbors', 'Logistic Regression', 'Decision Tree',
             'Random Forest','Easy Ensemble Classifier','AdaBoost␣
 ↪Classifier','SVM']
)
```



1.6 Conclusion

After analyzing the dataset using a variety of machine learning algorithms, the Random Forest model showed the best performance with an accuracy of 88% in predicting student dropout and academic success. Key predictors of success/dropout in graduation included factors such as Curricular units 2nd sem (grade), Curricular units 1st sem (grade), Course, Previous qualification,

Inflation rate, and GDP. The model valuable insights for early identification of at-risk students, which can help educational institutions design timely interventions. However, challenges such as data quality and feature imbalance were noted, and further work is needed to refine the model by incorporating additional features and addressing overfitting. Overall, this analysis provides actionable insights that can guide student retention strategies and improve academic success in educational institutions.