**Predict Students Dropout and Academic Success**

Challenges faced by students often follow common patterns. By identifying these shared obstacles, schools can address them more effectively, helping students achieve greater academic success.

Data source: <https://www.kaggle.com/datasets/syedfaizanalii/predict-students-dropout-and-academic-success/data>

**Cleaning**:

Clean the column names (specifically 'Daytime/evening attendance')

df.columns = df.columns.str.strip().str.replace('\t', '')  
df.columns = df.columns.str.replace('"', '')

Remove duplicates.

df.drop\_duplicates(inplace=True)

This project is analyzing whether they dropped out or graduated. If they’re still taking classes, their data is incomplete.

df = df[df['Target'] != 'Enrolled']

Change classified text under ‘Target’ column to be integers.

df.loc[:, 'Target'] = df['Target'].replace({'Dropout': 0, 'Graduate': 1})

Strip leading and trailing white-spaces.

df = df.map(lambda x: x.strip() if isinstance(x, str) else x)

Drop empty rows.

df.dropna(how='all', inplace=True)

Replace missing values with the mean.

df.fillna(df.mean(), inplace=True)

Set proper precision.

df = df.round(3)

**Spread**:

variance = df.var()  
std\_dev = df.std()  
q1 = df.quantile(0.25)  
q3 = df.quantile(0.75)  
iqr = q3 - q1  
  
print('Variance:')  
print(variance, '\n')  
print('Standard Deviation:')  
print(std\_dev, '\n')  
print('Inter-quartile Range (IQR):')  
print(iqr, '\n')

**Outliers** (using iqr from spread):

outliers = df[(df < (q1 - 1.5 \* iqr)) | (df > (q3 + 1.5 \* iqr))]  
print("Outliers:\n", outliers)  
outliers.to\_csv("outliers.xlsx", index=False)

*Outliers:*

*Marital status Application mode ... GDP Target*

*0 NaN NaN ... NaN NaN*

*1 NaN NaN ... NaN NaN*

*2 NaN NaN ... NaN NaN*

*3 NaN NaN ... NaN NaN*

*4 2.0 NaN ... NaN NaN*

*... ... ... ... ... ...*

*4419 NaN NaN ... NaN NaN*

*4420 NaN NaN ... NaN NaN*

*4421 NaN NaN ... NaN NaN*

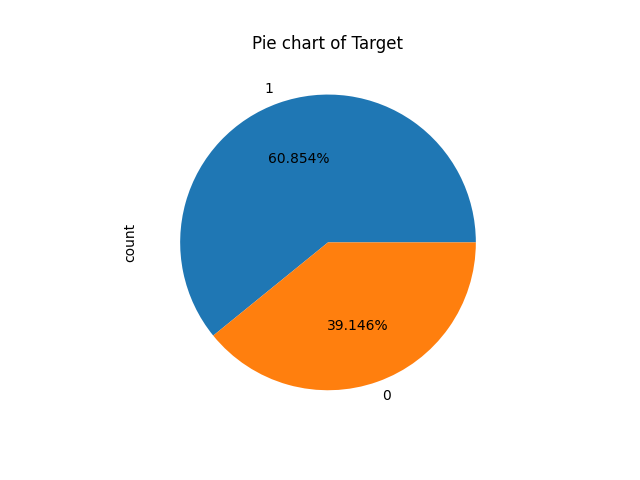
*4422 NaN NaN ... NaN NaN*

*4423 NaN NaN ... NaN NaN*

*[4424 rows x 37 columns]*

**Shape** of Target:

df['Target'].value\_counts().plot(kind='pie', autopct='%1.3f%%')  
plt.title('Pie chart of Target')  
plt.show()



Citing:

Random Forest

https://www.youtube.com/watch?app=desktop&v=\_QuGM\_FW9eo

https://www.kaggle.com/code/prashant111/random-forest-classifier-tutorial

[Linear Regression](https://www.geeksforgeeks.org/linear-regression-python-implementation/)

<https://www.geeksforgeeks.org/linear-regression-python-implementation/>

[Logistic Regression](https://www.datacamp.com/tutorial/understanding-logistic-regression-python)

<https://www.datacamp.com/tutorial/understanding-logistic-regression-python>

**Random Forest** is used because I need to see which columns matter more in predicting the target. The aim of this project was to not only predict whether a student will succeed but also analyze what factors need more attention.

**Logistic Regression** is used to predict which students graduate and which students drop out. It’s an algorithm that classifies students into a final binary value by analyzing many factors. In this case, it’s whether they will drop out of college or graduate college.

**KNN** is used to see the distance between the values. It gives an idea on the closeness of students’ profiles, which helps see if there are clusters of students that share common factors.

The program had great success in predicting the outcome of students. It was around a 90% success rate. ‘Curricular units 2nd sem (approved)' was the feature that mattered the most in a student’s outcome, and ‘International’ was the least important.