PREDICT STUDENT ACADEMIC SUCCESS OR DROPOUT

DATA DESCRIPTION:

	Marital status	Application mode	Application order	Course	Daytime/evening attendance\t	Previous qualification	Previous qualification (grade)	Nacionality	Mother's qualification	Father's qualification	...	Curricular units 2nd sem (credited)	Curricular units 2nd sem (enrolled)	Curricular units 2nd sem (evaluations)	Cu u (a
0	1	17	5.0	171.0	1.0	1.0	122.0	1.0	19.0	12.0	...	0	0	0	
1	1	15	1.0	9254.0	1.0	1.0	160.0	1.0	1.0	3.0	...	0	6	6	
2	1	1	5.0	9070.0	1.0	1.0	122.0	1.0	37.0	37.0	...	0	6	0	
3	1	17	2.0	9773.0	1.0	1.0	122.0	1.0	38.0	37.0	...	0	6	10	
4	2	39	1.0	8014.0	0.0	1.0	100.0	1.0	37.0	38.0	...	0	6	6	

5 rows × 37 columns

```
[4] df.shape # show the shape of the dataset
```

```
(4424, 37)
```

The pictures above describe the dataset. There are 4424 samples and 37 features before cleaning the data and dropping the null values. The data contains numerical and categorical values.

DATA PREPROCESSING:

```
# Drop the predictors for marital status,
df =df.drop(['Marital status', 'Nacionality', 'Application mode', 'Application order', 'GDP', 'Course'], axis = 1)
df.head() # show the dataset without unwanted features
```

	Daytime/evening attendance\t	Previous qualification	Previous qualification (grade)	Mother's qualification	Father's qualification	Mother's occupation	Father's occupation	Admission grade	Displaced	Educational special needs	...	Curricular units 1st sem (without evaluations)	Curricular units 2nd sem (credited)	Curricular units 2nd sem (enrolled)
0	1.0	1.0	122.0	19.0	12.0	5.0	9.0	127.3	1.0	0.0	...	0	0	0
1	1.0	1.0	160.0	1.0	3.0	3.0	3.0	142.5	1.0	0.0	...	0	0	6
2	1.0	1.0	122.0	37.0	37.0	9.0	9.0	124.8	1.0	0.0	...	0	0	6
3	1.0	1.0	122.0	38.0	37.0	5.0	3.0	119.6	1.0	0.0	...	0	0	6
4	0.0	1.0	100.0	37.0	38.0	9.0	9.0	141.5	0.0	0.0	...	0	0	6

5 rows × 31 columns

In the first picture you can see that after loading the data, I removed all of the unwanted features from the dataset. In this case, I dropped the features that I considered to be unimportant/indifferent to determine the success or dropout rate of a student (‘Marital status’, ‘Nationality’, ‘Application mode’, ‘Application order’,’GDP’, ‘Course’).

```
#check for null values
df.isnull().sum()
```

	0
Marital status	0
Application mode	0
Application order	1
Course	2
Daytime/evening attendance\t	3
Previous qualification	3
Previous qualification (grade)	5
Nacionality	5
Mother's qualification	4
Father's qualification	4
Mother's occupation	5
Father's occupation	4
Admission grade	3
Displaced	3
Educational special needs	3
Debtor	3
Tuition fees up to date	2
Gender	2
Scholarship holder	2
Age at enrollment	0

```
# Drop and check for null values
df.dropna(inplace = True)
df.isnull().sum()
```

	0
Marital status	0
Application mode	0
Application order	0
Course	0
Daytime/evening attendance\t	0
Previous qualification	0
Previous qualification (grade)	0
Nacionality	0
Mother's qualification	0
Father's qualification	0
Mother's occupation	0
Father's occupation	0
Admission grade	0
Displaced	0
Educational special needs	0
Debtor	0
Tuition fees up to date	0
Gender	0
Scholarship holder	0
Age at enrollment	0

```
df.shape[0] # number of samples
```

```
4412
```

```
df.shape[1] # find number of columns
```

```
37
```

The above images show the previous null values in the dataset and how I deleted them. After dropping the rows with null values I have 4412 samples and 37 features.

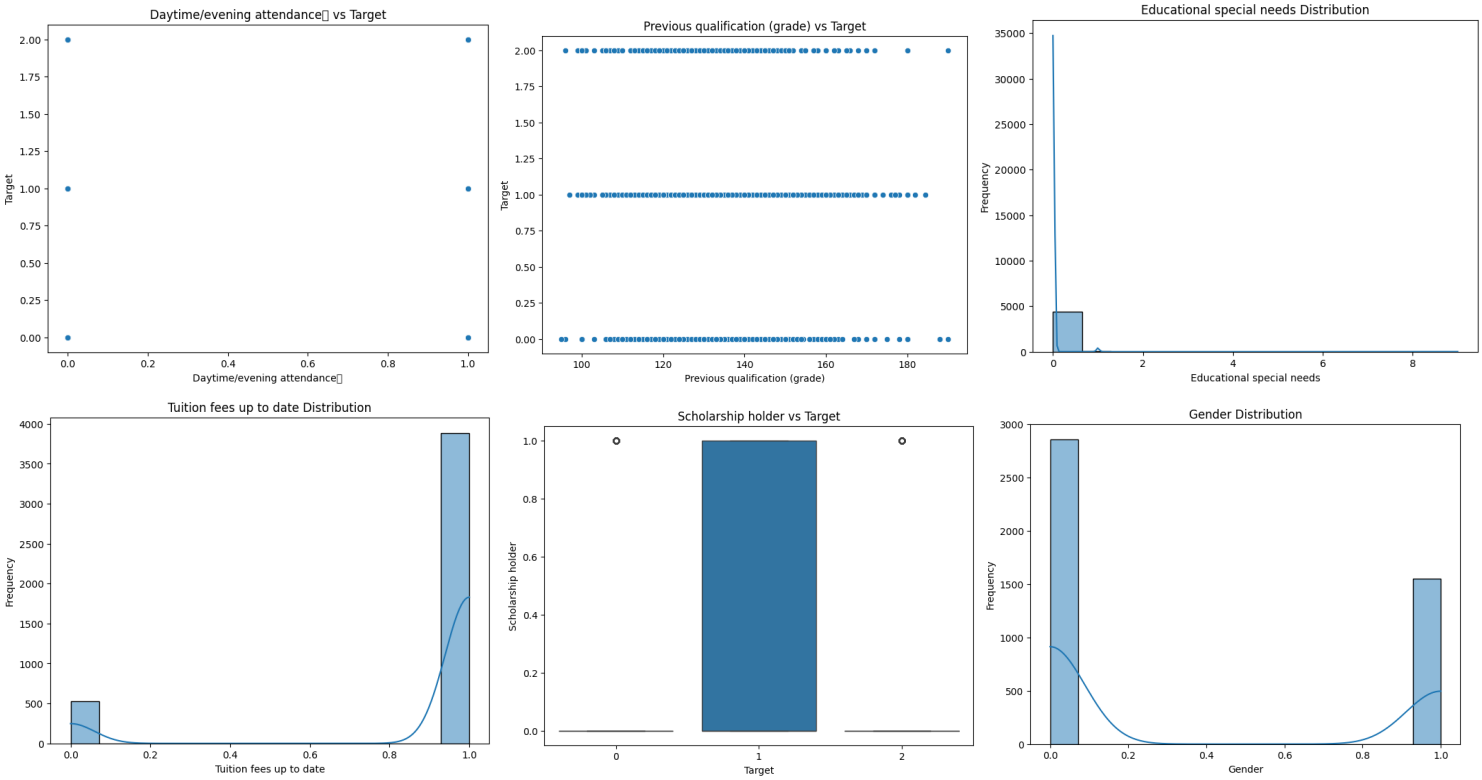
```
df.Target.unique() # Show unique values for Target
```

```
array(['Dropout', 'Graduate', 'Enrolled'], dtype=object)
```

```
df['Target'].replace(['Dropout','Graduate','Enrolled'], [0,1,2], inplace = True) # Replace values for Target
df.head() # show the values
```

In the previous image, I found the unique values for Target, which is my response value, and converted those categorical values into numerical ‘1,2,3’ to be able to categorize my findings.

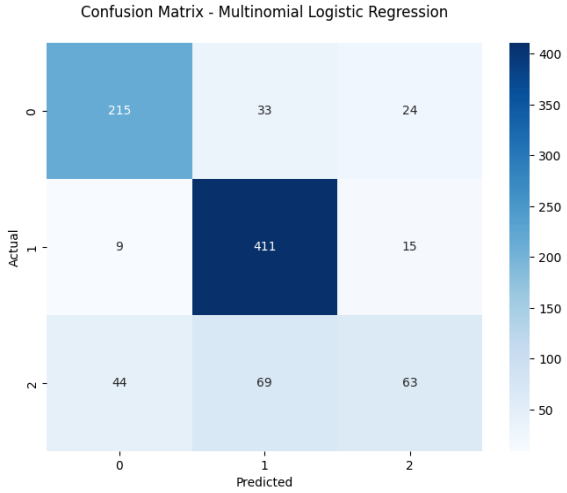
EXPLORATORY DATA ANALYSIS:



Based on the images above, we can see that some of the features are very significant to predict if a student will graduate or dropout, in other features we can just see the frequency of how many student there are, for example, we can see that there are more students with their tuition up to date and more female students than male students. Under these pictures, there are some positive and negative correlations between the Target and all the other features. A positive correlation suggests that higher values of a feature are associated with a higher likelihood of graduation.

MODEL DEVELOPMENT & PERFORMANCE EVALUATION:

MULTINOMIAL LOGISTIC REGRESSION:



Classification Report:

	precision	recall	f1-score	support
0	0.80	0.79	0.80	272
1	0.80	0.94	0.87	435
2	0.62	0.36	0.45	176
accuracy			0.78	883
macro avg	0.74	0.70	0.71	883
weighted avg	0.76	0.78	0.76	883

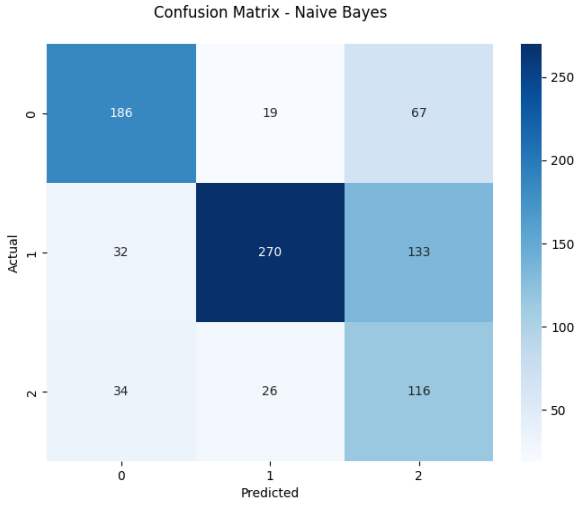
Accuracy: 0.78

When creating the Multinomial Logistic Regression model, I first splitted the data using 80% for training and 30% for testing with a random state of 42. Then I added a constant for the response variable of training and testing. To create the model I then used the Multinomial Logistic regression model with the dataset and the response value with the constant. The image below, with the table, shows us the relationship between the features and the response variable and how they are either positively and negatively correlated. After analyzing the table I predicted the target using the higher probability which then I used to calculate the confusion matrix.

The confusion matrix shows that there were 215 true predictions for dropout, 411 predictions for graduation, and 63 predictions for enrollment (which we are not really concerned about now). The classification report for Multinomial Logistic Regression shows an overall accuracy of 78% , a good precision (80%) for graduation, a great recall (94%) and f1-score (87%). For graduation and dropout, the model does not predict a lot of false wrong predictions.

Target=1	coef	std err	z	P>|z|	[0.025	0.975]
const	-2.1968	0.913	-2.407	0.016	-3.986	-0.408
Daytime/evening attendance	0.1316	0.233	0.566	0.572	-0.324	0.587
Previous qualification	0.0115	0.007	1.755	0.079	-0.001	0.024
Previous qualification (grade)	-0.0068	0.006	-1.137	0.256	-0.018	0.005
Mother's qualification	-0.0037	0.005	-0.718	0.473	-0.014	0.006
Father's qualification	0.0083	0.005	1.657	0.097	-0.002	0.018
Mother's occupation	0.0098	0.006	1.728	0.084	-0.001	0.021
Father's occupation	-0.0023	0.006	-0.372	0.710	-0.014	0.010
Admission grade	0.0084	0.006	1.509	0.131	-0.003	0.019
Displaced	-0.2429	0.143	-1.703	0.089	-0.522	0.037
Educational special needs	-0.2855	0.420	-0.679	0.497	-1.110	0.539
Debtor	-0.8218	0.231	-3.558	0.000	-1.274	-0.369
Tuition fees up to date	2.9838	0.299	9.985	0.000	2.398	3.570
Gender	-0.4097	0.136	-3.017	0.003	-0.676	-0.144
Scholarship holder	0.7734	0.167	4.632	0.000	0.446	1.101
Age at enrollment	-0.0449	0.011	-4.253	0.000	-0.066	-0.024
International	0.9946	0.432	2.303	0.021	0.148	1.841
Curricular units 1st sem (credited)	-0.1682	0.103	-1.632	0.103	-0.370	0.034
Curricular units 1st sem (enrolled)	-0.3055	0.133	-2.303	0.021	-0.566	-0.045
Curricular units 1st sem (evaluations)	-0.0168	0.034	-0.495	0.621	-0.083	0.050
Curricular units 1st sem (approved)	0.6829	0.076	8.966	0.000	0.534	0.832
Curricular units 1st sem (grade)	-0.1065	0.044	-2.420	0.016	-0.193	-0.020
Curricular units 1st sem (without evaluations)	0.1649	0.142	1.165	0.244	-0.113	0.442
Curricular units 2nd sem (credited)	-0.0835	0.111	-0.751	0.452	-0.301	0.134
Curricular units 2nd sem (enrolled)	-0.8718	0.127	-6.879	0.000	-1.120	-0.623
Curricular units 2nd sem (evaluations)	-0.0433	0.032	-1.345	0.179	-0.106	0.020
Curricular units 2nd sem (approved)	1.0366	0.072	14.360	0.000	0.895	1.178
Curricular units 2nd sem (grade)	0.1362	0.044	3.089	0.002	0.050	0.223
Curricular units 2nd sem (without evaluations)	0.1936	0.126	1.537	0.124	-0.053	0.440
Unemployment rate	2.953e-05	0.001	0.028	0.978	-0.002	0.002
Inflation rate	-0.0255	0.047	-0.549	0.583	-0.117	0.066

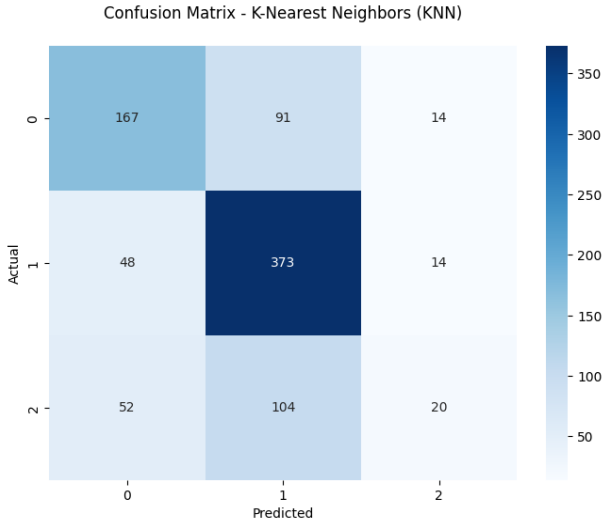
NAIVE BAYES:




```
Classification Report:
              precision    recall  f1-score   support

           0       0.74      0.68      0.71       272
           1       0.86      0.62      0.72       435
           2       0.37      0.66      0.47       176

    accuracy                           0.65       883
   macro avg       0.65      0.65      0.63       883
weighted avg       0.72      0.65      0.67       883

Accuracy: 0.65
```

For Naive Bayes I used the same training and testing but without adding the constant. I selected Naive Bayes because it is a good model for generalization, but it may not capture some differences. The confusion matrix shows good predictions for 186 dropouts, 270 for graduation, and 116 for enrollment. The classification report for Naive Bayes shows an overall accuracy of 65% , a good precision (86%) for graduation but a somewhat okay recall (62%) and f1-score (72%). In this case the Naive Bayes was less efficient than multinational logistic regression, except for the enrollment predictions.

KNN:




```
Classification Report:
              precision    recall  f1-score   support

           0       0.63      0.61      0.62       272
           1       0.66      0.86      0.74       435
           2       0.42      0.11      0.18       176

    accuracy                           0.63       883
   macro avg       0.57      0.53      0.51       883
weighted avg       0.60      0.63      0.59       883

Accuracy: 0.63
```

Only KNN has hyperparameters. I used thegridSearchCV which defines a parameter grid to find different n_neighbors and uses cross validation to find the best parameter. I also have the best_estimator_ for tuning and computing my analysis with the best neighbor. For the confusion matrix I used a heatmap for visualization.
The best neighbor was 5. For KNN I also used the same training and testing without constant values. KNN has good predictions of 167 dropouts, 373 graduations, and 20 enrollment. KNN performed better than Naive Bayes only in graduation predictions but it performed worse than Naive Bayes and Multinomial Logistic Regression for both dropouts and enrollment predictions. Classification report shows an overall accuracy of 63%, good recall of 86% and an okay precision (66%) and f1-scores (74%).

INTERPRETATION:

MODEL COMPARISON:

- Multinomial Logistic Regression performed better than both Naive Bayes and KNN. This could be due to the constant added to the fitting and predictions of the model. Other factors can be taken into account, for example, MLR is suited for multiclass problem, like in this cas, where a linear relationship exists between the features and the target variable; KNN can potentially capture great relationships in the data but might be prone to overfitting because it depends on the choice of number of neighbors and scaling features; and Naive Bayes is good for generalization but it might not capture some very small differences, it assumes feature independence which might not be valid for most cases.

SIGNIFICANT FEATURES:

After analyzing the table for Multinomial Logistic Regression, and looking at the coefficient and correlation between the features and the response value we have:

Positive Correlation (the higher the value the higher the likelihood of graduating): 'Tuitions up to date', 'International', 'Curricular units 1st sem (approved)', 'Curricular units 2nd sem (approved)', 'Curricular units 2nd sem (grade)'.

Negative Correlation (the lower the value, the lower the chance of graduating or higher changes of dripping out): 'Debtor', 'Gender', 'Age of enrollment', 'Curricular units 1st sem (enrolled)', 'Curricular units 1st sem (grade)', 'Curricular units 2nd sem (enrolled)'.

SPECIFIC CLUSTERS:

Logistic Regression: MLR classifies based on probabilities, as mentioned before higher correlation = higher probability of graduating; lower correlation = higher probability of dropout; average cases = enrolled.

KNN: clusters of similar features patterns lead to consistent classification.

Naive Bayes: Assigns probabilities based on feature likelihoods, as mentioned before, but often misclassified due to assumptions of feature independence.

RELATIONSHIP TO THE DATASET AND PROBLEM:

All these models aim to predict the graduation and dropout rates of students. Although, each model uses a different approach. For example, for MLR by examining the coefficient, one can identify the features that have strongest influence on the probability of the target; Naive Bayes assumes features independence, so the predictions are based on what each feature contributes to the model.; KNN relies on similarity between data points and it analyses the nearest neighbor for a given prediction.

SUMMARY:

In this particular case MLR outperforms because it balances precision, recall, and interpretability while handling the dataset's features. MLR has better outcomes and it is the model that better fits this task, this is due to the fact that it focuses on probabilities so it understands better why students succeed or fail. Naive Bayes struggles with feature dependencies, making this model unsuitable for this task. KNN is very sensitive to scaling but it provides reasonable results with specific tuning.

CONCLUSION & SELF REFLECTION:

In this project, I explored several classification machine learning models for predicting the graduation and dropout rates of students. To obtain great results one needs to clean the data and remove unnecessary features to accomplish a model that is trained with complete and concise data. I used hyperparameters for specific models and performed analysis on the confusion matrix, classification report, and summary of MLR.

I understood why we use different models for different problems and why it is important to understand how a model works.