CSE422 Lab Project Report

Predicting Academic Outcomes Using Machine Learning Models

Tasbir Ahmed
ID-22201028

Afra Musarrat Diya
ID-22201157

Group-1
Section 07

Table of contents

1.Introduction....3
2.Dataset Description....3
- Dataset Description.... 5
- Imbalanced Dataset....5
- Exploratory Data Analysis (EDA)....7

3. Dataset pre-processing....11
4. Dataset splitting....13
5. Model training & testing....14
Neural Network.... 14
Decision Tree....15
Logistic Regression.... 16
Kmeans:....17
6. Comparison analysis....18
7.Discussion and Conclusion:.... 19

1.Introduction

A key problem in educational data mining is predicting academic outcomes like graduation, dropout, or continuous enrollment. Universities that make reliable predictions are more likely to identify at risk students early on and take preventive measures to enhance support and retention programs.
The primary objective of this project is to predict students' academic outcomes based on the provided dataset.The project focuses on creating and evaluating machine learning models to divide students into three groups which are enrolled, graduate, and dropout on the basis of social, economic, academic, and cultural factors to identify at risk students and recognizing the key features that influence their success.Neural networks, Decision Tree, and Logistic Regression are among the classification models we have used to see which one works best. The study focuses on preparing the dataset through data preprocessing, handling class imbalance and evaluating the models using multiple performance metrics including accuracy, precision, recall, and F1-score to ensure a fair assessment. We also applied K-Means clustering to treat the problem as an unsupervised learning task and compared the discovered clusters with the actual academic outcomes.

2.Dataset Description

- **Dataset Description**
  - ➔ There are 24 features in the datasets. It has 4424 rows and 25 columns.

```
df.head()
```

	Marital status	Application mode	Application order	Course	Daytime/evening attendance\t	Previous qualification	Previous qualification (grade)	Nacionality	Mother's qualification	Father's qualification	...	Gender	Scholarship holder	Age at enrollment	International	Unemployment rate	Inflation rate	GDP	Target	Unnamed: 25	Unna
0	1.0	17.0	5.0	171.0	1.0	1.0	122.0	1.0	19.0	12.0	...	1.0	0.0	NaN	0.0	10.8	1.4	1.74	Dropout	NaN	
1	1.0	15.0	1.0	9254.0	1.0	1.0	160.0	1.0	1.0	3.0	...	1.0	0.0	19.0	0.0	13.9	-0.3	0.79	Graduate	NaN	
2	1.0	NaN	5.0	9070.0	1.0	1.0	122.0	NaN	37.0	37.0	...	1.0	NaN	19.0	0.0	10.8	1.4	1.74	Dropout	NaN	
3	1.0	17.0	2.0	9773.0	1.0	1.0	122.0	1.0	38.0	37.0	...	NaN	0.0	20.0	0.0	9.4	-0.8	-3.12	Graduate	NaN	
4	2.0	39.0	1.0	8014.0	0.0	1.0	100.0	1.0	37.0	NaN	...	0.0	0.0	NaN	0.0	13.9	-0.3	0.79	Graduate	NaN	

5 rows × 27 columns

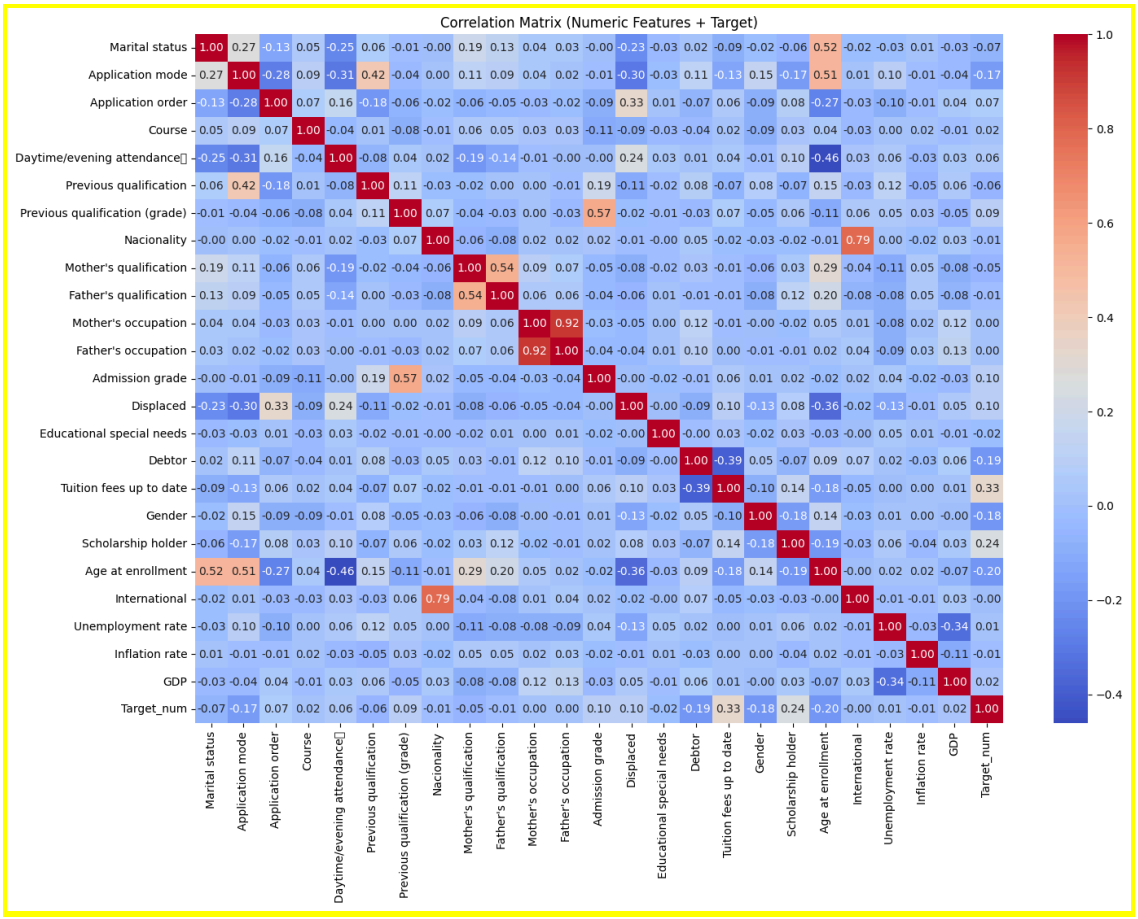
  - ➔ It is a classification problem. Here the 'Target' variable is categorical which means we have to predict a class. The samples in the dataset can be classified into 3 categories ('Dropout', 'Graduate', and 'Enrolled'). This categorical output makes it a classification problem.
  - ➔ The dataset contains a total of 4424 datapoints.
  - ➔ The dataset contains 7 quantitative and 17 categorical features
  - ➔ Yes,we need to encode categorical variables.Because models typically work with numerical data. So encoding helps to convert categorical data into numerical formats for the model to understand.

```
print('Shape of the dataset is {}. This dataset contains {} rows and {} columns.'.format(df.shape,df.shape[0],df.shape[1]))
```

```
Shape of the dataset is (4424, 25). This dataset contains 4424 rows and 25 columns.
```

```
[8] df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4424 entries, 0 to 4423
Data columns (total 25 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Marital status                  4001 non-null   float64
 1   Application mode                3941 non-null   float64
 2   Application order               3998 non-null   float64
 3   Course                          3959 non-null   float64
 4   Daytime/evening attendance      3984 non-null   float64
 5   Previous qualification          3990 non-null   float64
 6   Previous qualification (grade)  3952 non-null   float64
 7   Nacionality                     3978 non-null   float64
 8   Mother's qualification          4010 non-null   float64
 9   Father's qualification          3974 non-null   float64
 10  Mother's occupation             3988 non-null   float64
 11  Father's occupation             3999 non-null   float64
 12  Admission grade                 3962 non-null   float64
 13  Displaced                       3982 non-null   float64
 14  Educational special needs       3976 non-null   float64
 15  Debtor                          3972 non-null   float64
 16  Tuition fees up to date         3998 non-null   float64
 17  Gender                          3987 non-null   float64
 18  Scholarship holder              3974 non-null   float64
 19  Age at enrollment               3980 non-null   float64
 20  International                   3996 non-null   float64
 21  Unemployment rate               3995 non-null   float64
 22  Inflation rate                  4002 non-null   float64
 23  GDP                             3968 non-null   float64
 24  Target                          3971 non-null   object
dtypes: float64(24), object(1)
memory usage: 864.2+ KB
```



Correlation matrix analysis:

● Strong Redundancies:
  ○ Mother's vs Father's occupation (ρ ≈ 0.92) and mother's vs father's qualification (ρ ≈ 0.54–1.00) are mostly duplicate or same. We will now drop one column from each parental pair to avoid redundancy means duplicacy.

● Moderate Associations:
  ○ Admission grade - Previous qualification grade (ρ ≈ 0.57)
  ○ Age at enrollment - Application order (ρ ≈ 0.51)

● Weak or Negligible Links:
  ○ Most of the other feature–feature and feature–target correlations are almost zero, indicating It will help us to bring largely unique information.

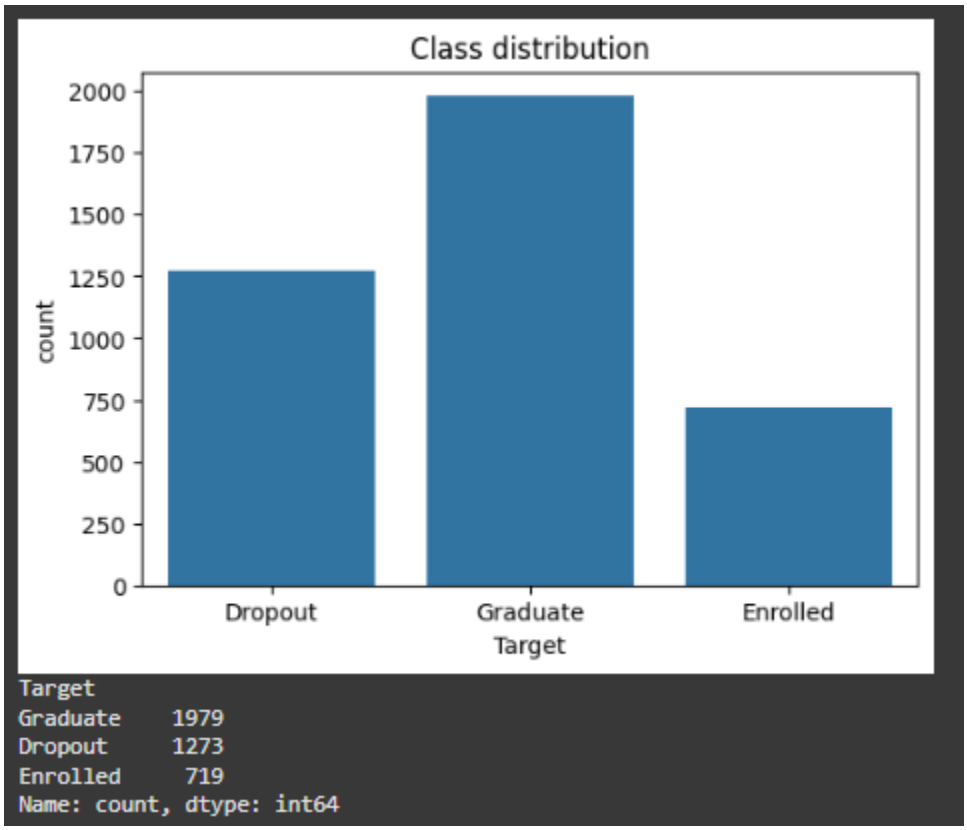
Based on the above analysis of correlation matrix, we eliminated the redundant parental columns but kept moderately correlated academic and demographic variables, and to maximize our prediction diversity, we have preserved all our other features.

- **Imbalanced Dataset**

	0
Marital status	423
Application mode	483
Application order	426
Course	465
Daytime/evening attendance\t	440
Previous qualification	434
Previous qualification (grade)	472
Nacionality	446
Mother's qualification	414
Father's qualification	450
Mother's occupation	436
Father's occupation	425
Admission grade	462
Displaced	442
Educational special needs	448
Debtor	452
Tuition fees up to date	426
Gender	437
Scholarship holder	450
Age at enrollment	444
International	428
Unemployment rate	429
Inflation rate	422
GDP	456
Target	453

dtype: int64

Here we can see the missing values or none values of each row. Basically it shows how many missing values are there for each feature as well as columns including target.




```
Target
Graduate    1979
Dropout     1273
Enrolled     719
Name: count, dtype: int64
```

From the following bar chart, It is clear that the unique classes do not have an unique number of instances . As the number of samples for each unique category is not equal, it is an imbalanced dataset. (Graduate>Dropout>Enrolled) (unclassified 453)

- **Exploratory Data Analysis (EDA)**

  ➔ Descriptive stats

```
1  display(df.describe(include='all').T)
```

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Marital status	4001.0	NaN	NaN	NaN	1.172457	0.584595	1.0	1.0	1.0	1.0	6.0
Application mode	3941.0	NaN	NaN	NaN	18.56204	17.453216	1.0	1.0	17.0	39.0	57.0
Application order	3998.0	NaN	NaN	NaN	1.730865	1.310683	0.0	1.0	1.0	2.0	9.0
Course	3959.0	NaN	NaN	NaN	8856.628947	2063.151599	33.0	9085.0	9238.0	9556.0	9991.0
Daytime/evening attendance\t	3984.0	NaN	NaN	NaN	0.892319	0.310016	0.0	1.0	1.0	1.0	1.0
Previous qualification	3990.0	NaN	NaN	NaN	4.525564	10.132748	1.0	1.0	1.0	1.0	43.0
Previous qualification (grade)	3952.0	NaN	NaN	NaN	132.690056	13.226582	95.0	125.0	133.1	140.0	190.0
Nacionality	3978.0	NaN	NaN	NaN	1.860985	7.040422	1.0	1.0	1.0	1.0	109.0
Mother's qualification	4010.0	NaN	NaN	NaN	19.482294	15.620541	1.0	2.0	19.0	37.0	44.0
Father's qualification	3974.0	NaN	NaN	NaN	22.20307	15.382146	1.0	3.0	19.0	37.0	44.0
Mother's occupation	3988.0	NaN	NaN	NaN	10.895687	26.304799	0.0	4.0	5.0	9.0	194.0
Father's occupation	3999.0	NaN	NaN	NaN	10.910978	25.003303	0.0	4.0	7.0	9.0	194.0
Admission grade	3962.0	NaN	NaN	NaN	126.903609	14.506892	95.0	117.8	126.1	134.6	190.0
Displaced	3982.0	NaN	NaN	NaN	0.550477	0.497508	0.0	0.0	1.0	1.0	1.0
Educational special needs	3976.0	NaN	NaN	NaN	0.011569	0.106951	0.0	0.0	0.0	0.0	1.0
Debtor	3972.0	NaN	NaN	NaN	0.114804	0.318825	0.0	0.0	0.0	0.0	1.0
Tuition fees up to date	3998.0	NaN	NaN	NaN	0.882941	0.32153	0.0	1.0	1.0	1.0	1.0
Gender	3987.0	NaN	NaN	NaN	0.352646	0.477854	0.0	0.0	0.0	1.0	1.0
Scholarship holder	3974.0	NaN	NaN	NaN	0.251132	0.433719	0.0	0.0	0.0	1.0	1.0
Age at enrollment	3980.0	NaN	NaN	NaN	23.227638	7.608241	17.0	19.0	20.0	25.0	70.0
International	3996.0	NaN	NaN	NaN	0.025526	0.157735	0.0	0.0	0.0	0.0	1.0
Unemployment rate	3995.0	NaN	NaN	NaN	11.575394	2.665618	7.6	9.4	11.1	13.9	16.2
Inflation rate	4002.0	NaN	NaN	NaN	1.228286	1.380863	-0.8	0.3	1.4	2.6	3.7
GDP	3968.0	NaN	NaN	NaN	0.017656	2.274384	-4.06	-1.7	0.32	1.79	3.51
Target	3971	3	Graduate	1979	NaN	NaN	NaN	NaN	NaN	NaN	NaN

➔ Summery statics of numerical features

	count	mean	std	min	25%	50%	75%	max
Application order	3998.0	1.730865	1.310683	0.00	1.0	1.00	2.00	9.00
Previous qualification (grade)	3952.0	132.690056	13.226582	95.00	125.0	133.10	140.00	190.00
Admission grade	3962.0	126.903609	14.506892	95.00	117.8	126.10	134.60	190.00
Age at enrollment	3980.0	23.227638	7.608241	17.00	19.0	20.00	25.00	70.00
Unemployment rate	3995.0	11.575394	2.665618	7.60	9.4	11.10	13.90	16.20
Inflation rate	4002.0	1.228286	1.380863	-0.80	0.3	1.40	2.60	3.70
GDP	3968.0	0.017656	2.274384	-4.06	-1.7	0.32	1.79	3.51

```
1  numerical_data.var()
```

	0
Application order	1.717891
Previous qualification (grade)	174.942475
Admission grade	210.449911
Age at enrollment	57.885338
Unemployment rate	7.105521
Inflation rate	1.906783
GDP	5.172822

Variance

```
1  numerical_data.skew()
```

	0
Application order	1.872101
Previous qualification (grade)	0.307966
Admission grade	0.515351
Age at enrollment	2.091374
Unemployment rate	0.207260
Inflation rate	0.255016
GDP	-0.400097

Skew

➔ Skewness Interpretation

- Application order (1.871): There is a significant right-skew, with some students having higher application order numbers but the majority have lower numbers.
- Previous qualification grade (0.307): Fairly symmetrical – slight positive skew, close to normal distribution.
- Admission grade (0.499): Moderately right-skewed; some students have significantly higher grades than the mean, but more students have grades below it.
- Age at enrollment (2.074): Strong right-skew – most students are younger; some older students create a long right tail.
- The unemployment rate (0.210) is near to normal and has a somewhat right-skewed distribution.
- Inflation rate (0.262): Slightly right-skewed – fairly balanced distribution with minor positive skew.
- GDP (-0.407): Moderately left-skewed, with more observations above the mean; the GDP numbers of several nations are noticeably lower.

➔ Summary Statistics of Categorical Features

	count	mean	std	min	25%	50%	75%	max
Marital status	4001.0	1.172457	0.584595	1.0	1.0	1.0	1.0	6.0
Application mode	3941.0	18.562040	17.453216	1.0	1.0	17.0	39.0	57.0
Course	3959.0	8856.628947	2063.151599	33.0	9085.0	9238.0	9556.0	9991.0
Daytime/evening attendance\t	3984.0	0.892319	0.310016	0.0	1.0	1.0	1.0	1.0
Previous qualification	3990.0	4.525564	10.132748	1.0	1.0	1.0	1.0	43.0
Nacionality	3978.0	1.860985	7.040422	1.0	1.0	1.0	1.0	109.0
Mother's qualification	4010.0	19.482294	15.620541	1.0	2.0	19.0	37.0	44.0
Father's qualification	3974.0	22.203070	15.382146	1.0	3.0	19.0	37.0	44.0
Mother's occupation	3988.0	10.895687	26.304799	0.0	4.0	5.0	9.0	194.0
Father's occupation	3999.0	10.910978	25.003303	0.0	4.0	7.0	9.0	194.0
Displaced	3982.0	0.550477	0.497508	0.0	0.0	1.0	1.0	1.0
Educational special needs	3976.0	0.011569	0.106951	0.0	0.0	0.0	0.0	1.0
Debtor	3972.0	0.114804	0.318825	0.0	0.0	0.0	0.0	1.0
Tuition fees up to date	3998.0	0.882941	0.321530	0.0	1.0	1.0	1.0	1.0
Gender	3987.0	0.352646	0.477854	0.0	0.0	0.0	1.0	1.0
Scholarship holder	3974.0	0.251132	0.433719	0.0	0.0	0.0	1.0	1.0
International	3996.0	0.025526	0.157735	0.0	0.0	0.0	0.0	1.0

```
1  categorical_data.var()
```

	0
Marital status	3.417512e-01
Application mode	3.046147e+02
Course	4.256595e+06
Daytime/evening attendance\t	9.610971e-02
Previous qualification	1.026726e+02
Nacionality	4.956754e+01
Mother's qualification	2.440013e+02
Father's qualification	2.366104e+02
Mother's occupation	6.919425e+02
Father's occupation	6.251652e+02
Displaced	2.475142e-01
Educational special needs	1.143844e-02
Debtor	1.016493e-01
Tuition fees up to date	1.033817e-01
Gender	2.283441e-01
Scholarship holder	1.881122e-01
International	2.488020e-02

```
1  categorical_data.skew()
```

	0
Marital status	4.429740
Application mode	0.398201
Course	-3.808295
Daytime/evening attendance\t	-2.532235
Previous qualification	2.900447
Nacionality	10.924248
Mother's qualification	0.010805
Father's qualification	-0.291581
Mother's occupation	5.383512
Father's occupation	5.488079
Displaced	-0.203022
Educational special needs	9.138353
Debtor	2.417568
Tuition fees up to date	-2.383186
Gender	0.617042
Scholarship holder	1.148176
International	6.019125

```
1  categorical_data.nunique()
```

	0
Marital status	6
Application mode	18
Course	17
Daytime/evening attendance\t	2
Previous qualification	17
Nacionality	20
Mother's qualification	28
Father's qualification	34
Mother's occupation	32
Father's occupation	43
Displaced	2
Educational special needs	2
Debtor	2
Tuition fees up to date	2
Gender	2
Scholarship holder	2
International	2

Distribution and outer analysis:

The dataset showed some imbalances and mild outliers.Most features looked normal, but a few needed adjustments. Binary features (like Debtor, Scholarship holder) were mostly one-sided, some rare categories were grouped as “Other,” and grades were slightly skewed but fixed with scaling. In the age group,there are many young students but also some older ones. Economic features (like GDP, inflation) were used as numbers or grouped.Overall, no severe outliers were found, and preprocessing steps ensured stable modeling.

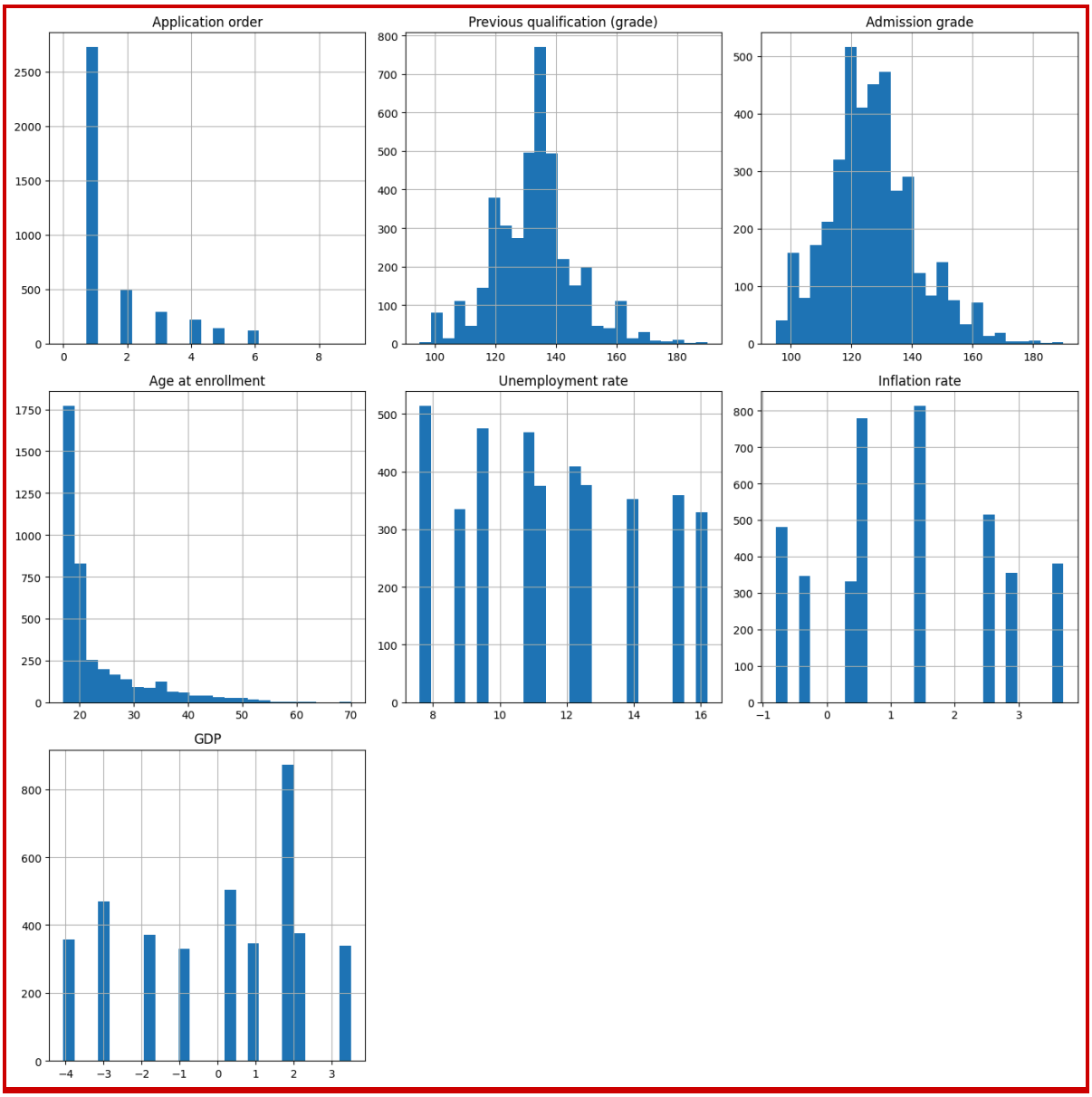
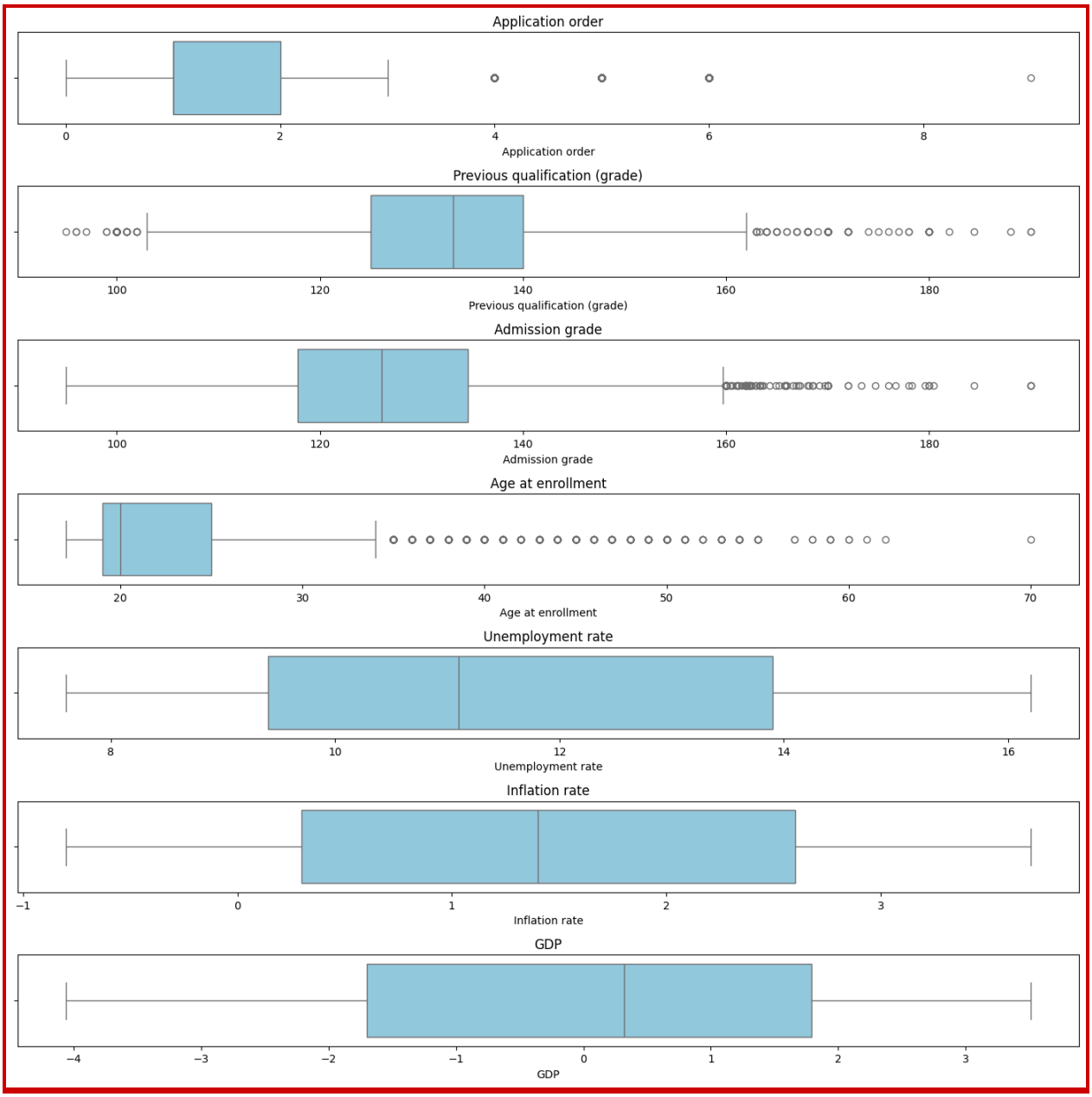


Figure: Box Plot



3. Dataset pre-processing

Fault1: Missing Target Labels/Missing value:

percentage of missing values in target column: (453/4424)*100=10.24 %

About 10.24 % of the rows lack a value for the Target (Graduate / Dropout / Enrolled). Without a ground-truth label, these examples cannot be used to train or evaluate our supervised models.

To solve the issue we dropped all rows where Target is null.This removal ensures that every remaining example contributes valid feedback during model training and evaluation.

```
# ▶ drop rows with missing Target
df = df.dropna(subset=['Target']).reset_index(drop=True)
```

Fault 2: Irrelevant or Redundant Features:

Correlation analysis revealed several numeric-coded columns that had zero or near-zero correlation with the target. Keeping these columns would only add noise, slow training, and for linear models inject multicollinearity without predictive gain.

To solve this problem we dropped these four columns outright before any further processing:

```
drop_col = ['Target',"Father's qualification","Father's occupation","Mother's occupation"]
```

Fault 3: Mixed Data Types & Categorical Codes

The dataset contains several features that are categorical or binary flags(Yes/No) but represented as Integers.If left numeric,models may treat them as continuously ordered or produce meaningless distance calculations.

To solve this problem, we have manually designated each predictor as either categorical or numeric, based on domain knowledge. Then In our preprocessing pipeline, categorical features are imputed with the most frequent value and then one-hot encoded. Whereas numeric features are imputed with the median and standardized to zero mean/unit variance.

```
drop_col = ['Target',"Father's qualification","Father's occupation","Mother's occupation"]

cat_feats = [
    "Marital status",
    "Application mode",
    "Course",
    "Daytime/evening attendance\t",
    "Previous qualification",
    "Nacionality",
    "Mother's qualification",
    # "Father's qualification",
    # "Mother's occupation",
    # "Father's occupation",
    "Displaced",
    "Educational special needs",
    "Debtor",
    "Tuition fees up to date",
    "Gender",
    "Scholarship holder",
    "International"
]

num_feats = [
    "Application order",
    "Previous qualification (grade)",
    "Admission grade",
    "Age at enrollment",
    "Unemployment rate",
    "Inflation rate",
    "GDP"
]
```

Fault 4: Missing Values in Predictors
Every predictor column has roughly 9–11 % missing values. Imputing naively (dropping rows) would discard too much data.

To solve this problem, for the numeric pipeline we used median imputation.Median imputation is robust to outliers and preserves the relative ranking of exam scores, ages, and economic indicators.

```
numeric_pipeline = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='median')),
    ('scaler',  StandardScaler())
])
```

For the categorical pipeline,we used mode imputation and one hot encoding. Mode imputation preserves the dominant category and one-hot encoding avoids imposing artificial ordinal scales on the codes.

```
categorical_pipeline = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='most_frequent')),
    ('onehot',  OneHotEncoder(handle_unknown='ignore'))
])
```

4. Dataset splitting

We first split the cleaned dataset into training (70 %) and test (30 %) sets, stratified by Target.Then we fit our ColumnTransformer only on X_train, and transform both X_train and X_test using that fitted preprocessor. This guarantees that no statistics from the test set influences the training pipeline.

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, stratify=y, random_state=42)
```

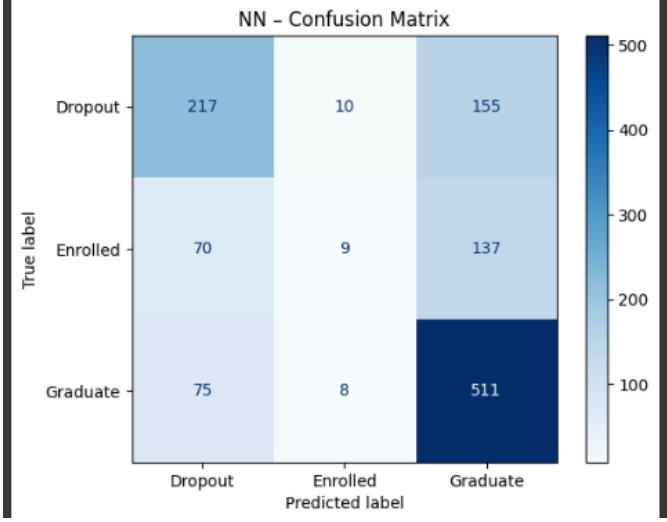
```
preprocessor = ColumnTransformer(
    transformers=[
        ('num', numeric_pipeline, num_feats),
        ('cat', categorical_pipeline, cat_feats)
    ])
```

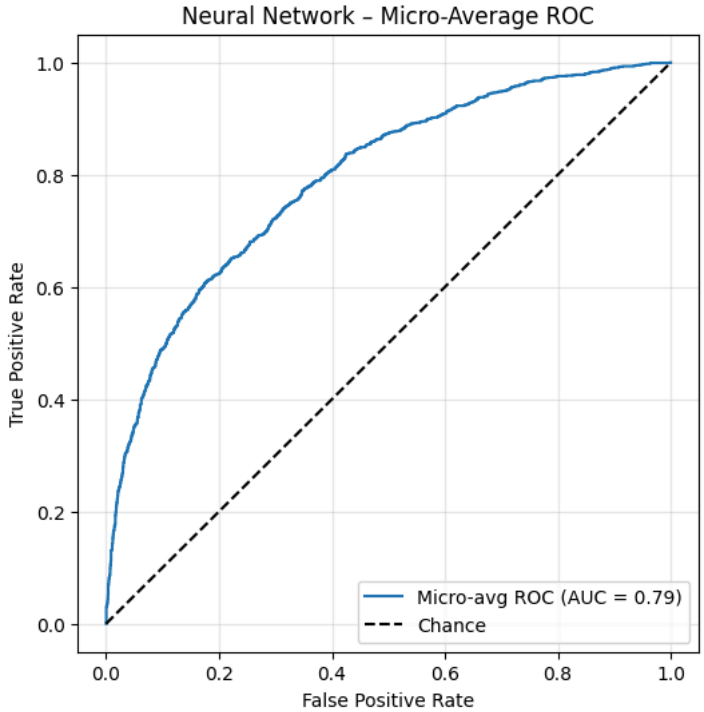
5. Model training & testing

Neural Network

Neural Network

	precision	recall	f1-score	support
Dropout	0.60	0.57	0.58	382
Enrolled	0.33	0.04	0.07	216
Graduate	0.64	0.86	0.73	594
accuracy			0.62	1192
macro avg	0.52	0.49	0.46	1192
weighted avg	0.57	0.62	0.56	1192

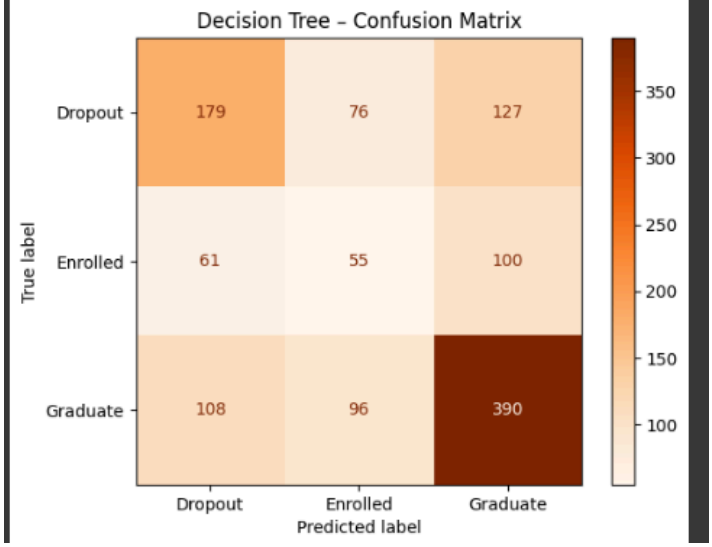


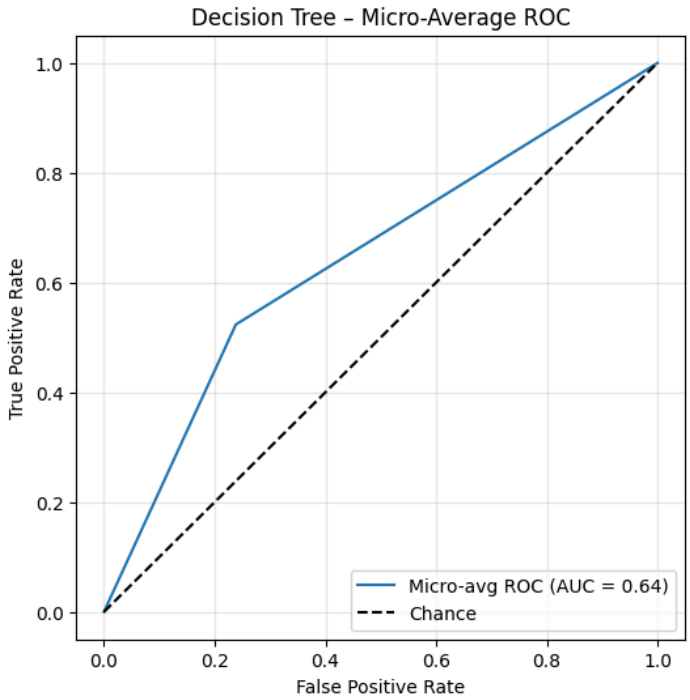


Decision Tree

Decision Tree

	precision	recall	f1-score	support
Dropout	0.51	0.47	0.49	382
Enrolled	0.24	0.25	0.25	216
Graduate	0.63	0.66	0.64	594
accuracy			0.52	1192
macro avg	0.46	0.46	0.46	1192
weighted avg	0.52	0.52	0.52	1192

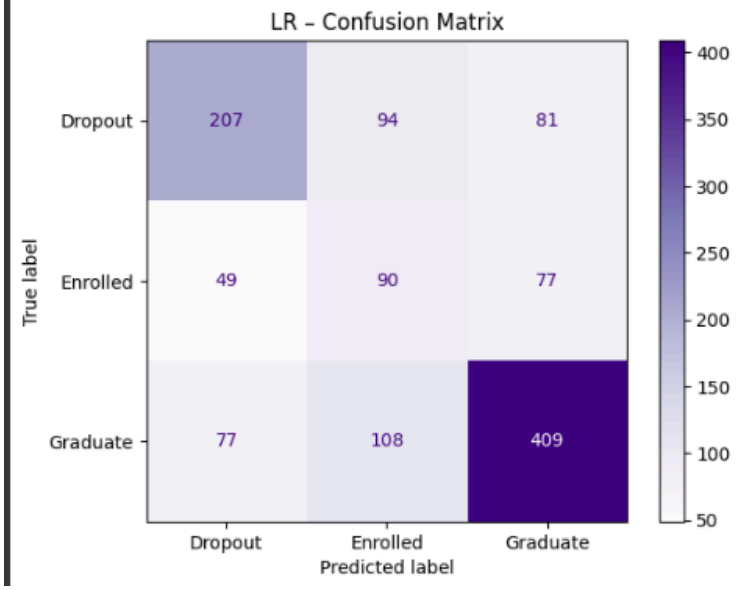


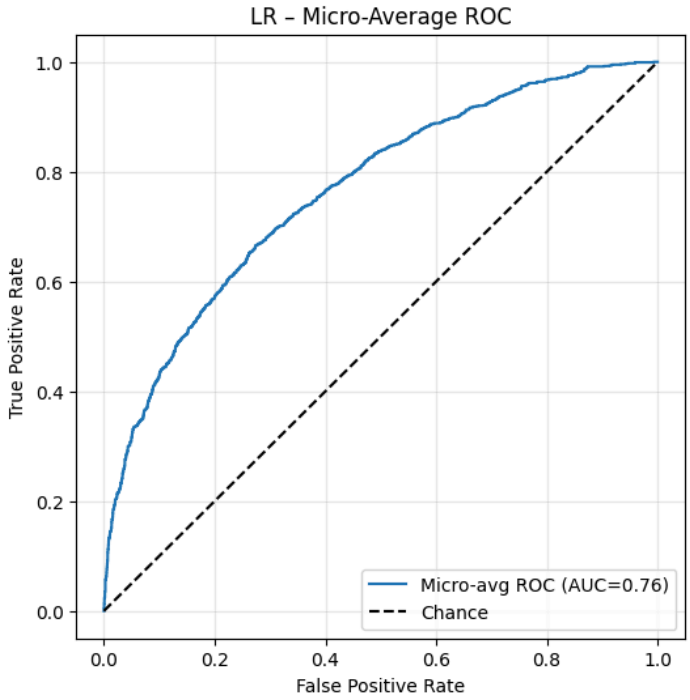


Logistic Regression

Logistic Regression

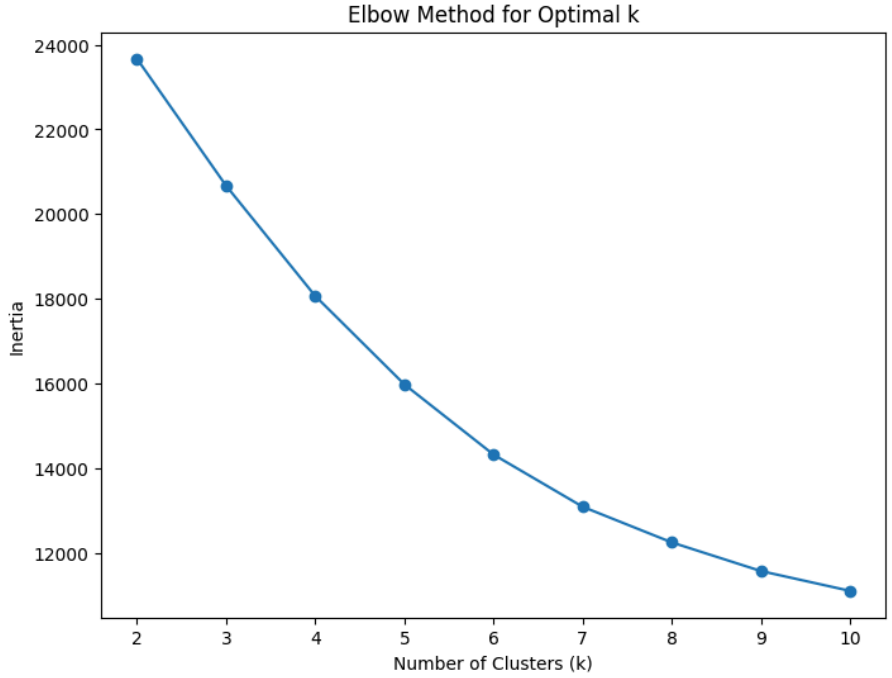
	precision	recall	f1-score	support
Dropout	0.62	0.54	0.58	382
Enrolled	0.31	0.42	0.35	216
Graduate	0.72	0.69	0.70	594
accuracy			0.59	1192
macro avg	0.55	0.55	0.55	1192
weighted avg	0.61	0.59	0.60	1192

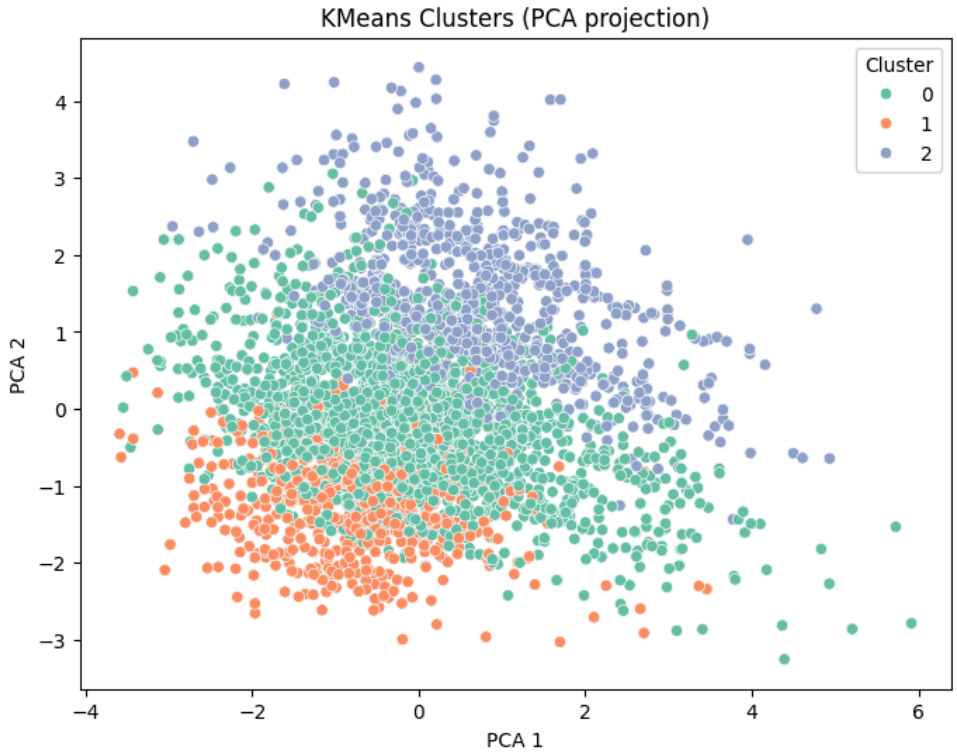




Kmeans:

We are treating this problem also as an unsupervised learning problem and then applying k means clustering;





Cluster	Application order	Previous qualification (grade)	Admission grade	V
0	1.169919	132.396997	127.178695	
1	4.224329	131.317491	124.650811	
2	1.272834	134.179553	127.518888	

Cluster	Age at enrollment	Unemployment rate	Inflation rate	GDP
0	23.894540	10.454767	0.808525	0.846538
1	19.243007	10.728142	1.047803	0.542791
2	24.350816	14.753975	2.402442	-2.320965

6. Comparison analysis

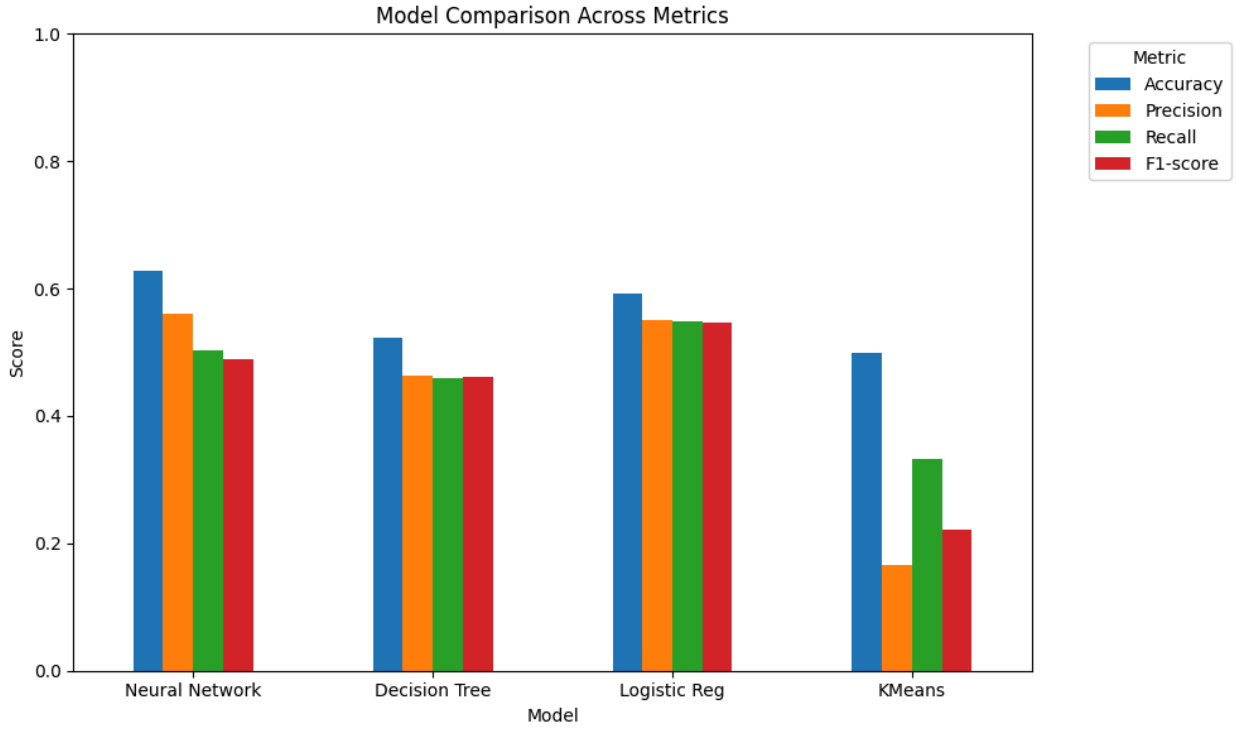


Figure: Comparison of classification metrics of the models

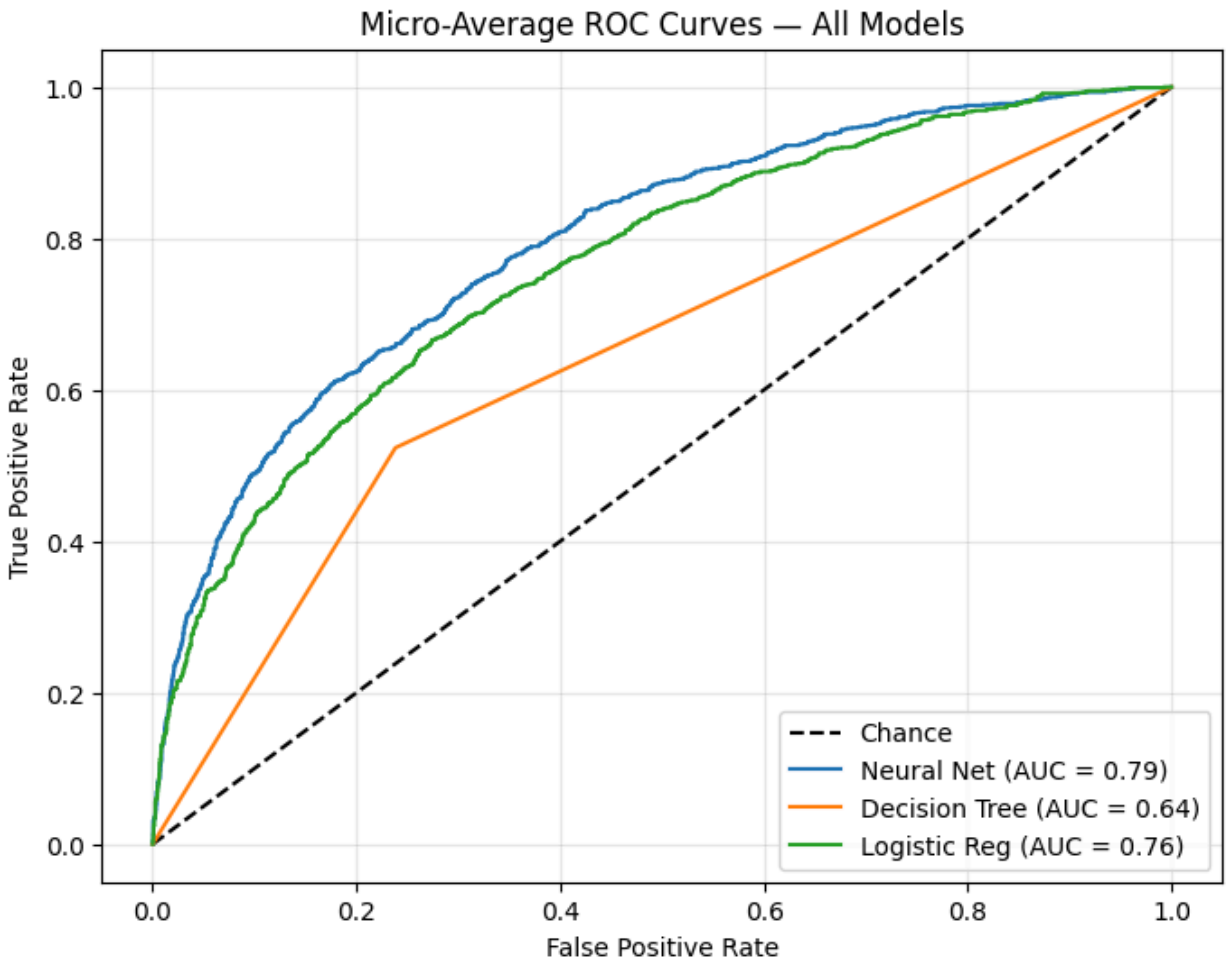


Figure: ROC Curve

7.Discussion and Conclusion:

Understanding the Results:
The performance of all models such as Neural Network, Decision Tree, and Logistic Regression gives us an overall accuracy ranging from 52% (Decision Tree) to 62% (Neural Network). Precision, recall, and F1-scores shows notable imbalance across classes.

● Neural Network performs 62% in overall accuracy but struggles to recall the 'Enrolled' class, likely due to its limited representation in the training set and a lack of oversampling strategies.

● Logistic Regression offers well performance, especially in terms of micro and weighted averages, suggesting that linear relationships contribute meaningfully to this classification task.

● Decision Tree underperformed, with lower accuracy and micro-average scores, Because of their sensitivity to high-dimensional data and class imbalance.

● From the above graph we are seeing that after applying kmeans clustering it's performance is (As we are treating this problem also as an unsupervised learning problem,) lower than other just because it is unsupervised learning.

Causes of These Findings

1. Class Imbalance: The "Enrolled" class receives the least amount of funding (around 18%), and this disparity is not specifically addressed. Models are therefore skewed toward forecasting "Graduate" and "Dropout," which are more common in the dataset.

2. Feature Informativeness: Tree-based models that can effectively use these characteristics benefit from the favorable correlation between the target and tuition fee status and admission grades.Nevertheless, the discriminative power of the model is diminished when the signal in certain categorical variables is smaller.

3. Redundant Features: Multicollinearity was introduced by a very high correlation between parental occupation and qualification. Model stability increased after these were removed, but performance was still constrained by intrinsic noise and data quality.

4. Enrolled Class Detection: All models have poor recall on the "Enrolled" class, most likely as a result of their feature space overlap with "Dropout" or "Graduate," which makes it difficult to distinguish with confidence.

Difficulties in the Modeling Process:

● Severe Class Imbalance: Poor generalization for this class across all models is caused by the tiny percentage of "Enrolled" cases.

● Neural Network Instability: Despite their good performance, neural networks were vulnerable to early pausing and dropout rates during training. Due to a lack of specific sampling or loss algorithms, they also performed worse on minority classes.

In this overall process neural network performed well enough and gives us better results than other models