



دانشگاه اصفهان
دانشکده مهندسی کامپیوتر

هوش محاسباتی
گزارش کتبی تمرین دوم
بخش چهارم: پروژه نهایی

اعضای گروه:

رضا چراخ

فرناز موحدی

اردیبهشت ۱۴۰۴

فهرست مطالب

صفحه

عنوان

2	فصل اول پیاده‌سازی پروژه و تحلیل نتایج
2	۱-۱- مقدمه
2	۱-۲- پیش‌پردازش و اکتشاف داده‌ها
6	۱-۳- فازی سازی ویژگی‌ها
8	۱-۴- استخراج قوانین فازی (روش Mendel-Wang)
10	۱-۵- انتخاب قوانین با استفاده از الگوریتم ژنتیک
12	۱-۶- استنتاج فازی برای طبقه بندی
12	۱-۷- ارزیابی مدل
14	منابع

فصل اول

پیاده‌سازی پروژه و تحلیل نتایج

۱-۱- مقدمه

۱-۲- پیش‌پردازش و اکتشاف داده‌ها

با بررسی دیتاست متوجه شدیم که مقادیر تمامی ستون‌ها (۳۶ ستون) به جز ستون هدف (Target)، از جمله ویژگی‌های طبقه‌ای عددی هستند یا تبدیل به مقادیر عددی شده‌اند.

```
RangeIndex: 4424 entries, 0 to 4423
Data columns (total 37 columns):
 #   Column                                                                 Non-Null Count  Dtype
---  -
 0   Marital status                                                         4424 non-null   int64
 1   Application mode                                                         4424 non-null   int64
 2   Application order                                                         4424 non-null   int64
 3   Course                                                                  4424 non-null   int64
 4   Daytime/evening attendance                                              4424 non-null   int64
 5   Previous qualification                                                  4424 non-null   int64
 6   Previous qualification (grade)                                          4424 non-null   float64
 7   Nacionality                                                            4424 non-null   int64
 8   Mother's qualification                                                  4424 non-null   int64
 9   Father's qualification                                                  4424 non-null   int64
10   Mother's occupation                                                     4424 non-null   int64
11   Father's occupation                                                     4424 non-null   int64
12   Admission grade                                                         4424 non-null   float64
13   Displaced                                                              4424 non-null   int64
14   Educational special needs                                              4424 non-null   int64
15   Debtor                                                                  4424 non-null   int64
16   Tuition fees up to date                                                4424 non-null   int64
17   Gender                                                                  4424 non-null   int64
18   Scholarship holder                                                     4424 non-null   int64
19   Age at enrollment                                                      4424 non-null   int64
20   International                                                           4424 non-null   int64
21   Curricular units 1st sem (credited)                                    4424 non-null   int64
22   Curricular units 1st sem (enrolled)                                    4424 non-null   int64
23   Curricular units 1st sem (evaluations)                                4424 non-null   int64
24   Curricular units 1st sem (approved)                                    4424 non-null   int64
25   Curricular units 1st sem (grade)                                        4424 non-null   float64
26   Curricular units 1st sem (without evaluations)                        4424 non-null   int64
27   Curricular units 2nd sem (credited)                                    4424 non-null   int64
28   Curricular units 2nd sem (enrolled)                                    4424 non-null   int64
29   Curricular units 2nd sem (evaluations)                                4424 non-null   int64
30   Curricular units 2nd sem (approved)                                    4424 non-null   int64
31   Curricular units 2nd sem (grade)                                        4424 non-null   float64
32   Curricular units 2nd sem (without evaluations)                        4424 non-null   int64
33   Unemployment rate                                                      4424 non-null   float64
34   Inflation rate                                                         4424 non-null   float64
35   GDP                                                                    4424 non-null   float64
36   Target                                                                4424 non-null   object
```

همچنین مشاهده شد که هیچ مقدار گم شده‌ای در دیتاست وجود ندارد.

```
print(df.isnull().sum())
```

```
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
International       0
Curricular units 1st sem (credited) 0
Curricular units 1st sem (enrolled) 0
Curricular units 1st sem (evaluations) 0
Curricular units 1st sem (approved) 0
Curricular units 1st sem (grade) 0
Curricular units 1st sem (without evaluations) 0
Curricular units 2nd sem (credited) 0
Curricular units 2nd sem (enrolled) 0
Curricular units 2nd sem (evaluations) 0
Curricular units 2nd sem (approved) 0
Curricular units 2nd sem (grade) 0
Curricular units 2nd sem (without evaluations) 0
Unemployment rate   0
Inflation rate      0
GDP                 0
Target              0
dtype: int64
```

تعداد مقادیر منحصر به فرد ویژگی‌ها هم نشان داد تعدادی ویژگی باینری در بین بقیه ویژگی‌ها هم

وجود دارد.

Marital status	6
Application mode	18
Application order	8
Course	17
Daytime/evening attendance\t	2
Previous qualification	17
Previous qualification (grade)	101
Nacionality	21
Mother's qualification	29
Father's qualification	34
Mother's occupation	32
Father's occupation	46
Admission grade	620
Displaced	2
Educational special needs	2
Debtor	2
Tuition fees up to date	2
Gender	2
Scholarship holder	2
Age at enrollment	46
International	2
Curricular units 1st sem (credited)	21
Curricular units 1st sem (enrolled)	23
Curricular units 1st sem (evaluations)	35
Curricular units 1st sem (approved)	23
Curricular units 1st sem (grade)	805
Curricular units 1st sem (without evaluations)	11
Curricular units 2nd sem (credited)	19
Curricular units 2nd sem (enrolled)	22
Curricular units 2nd sem (evaluations)	30
Curricular units 2nd sem (approved)	20
Curricular units 2nd sem (grade)	786
Curricular units 2nd sem (without evaluations)	10
Unemployment rate	10
Inflation rate	9
GDP	10
Target	3

dtype: int64

برای پیش پردازش داده‌ها ابتدا با استفاده از one-hot-encoding ویژگی Target را به سه ویژگی باینری Target_Dropout, Target_Enrolled و Target_Graduate تبدیل کردیم.

```
df = pd.get_dummies(df, columns=["Target"], prefix="Target")
```

+ Code + Markdown

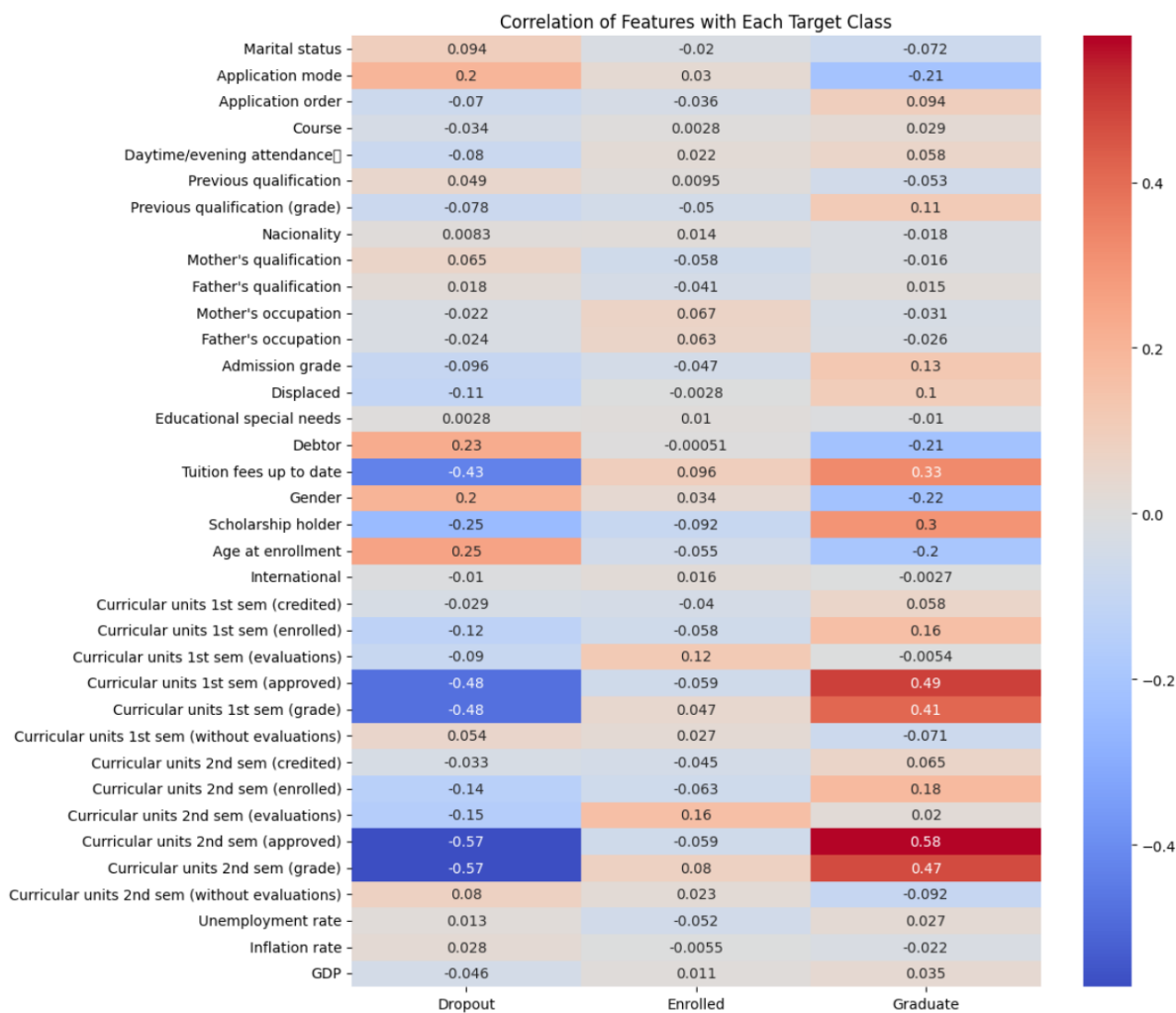
```
df["Target_Dropout"] = df["Target_Dropout"].astype(int)
```

```
df["Target_Enrolled"] = df["Target_Enrolled"].astype(int)
```

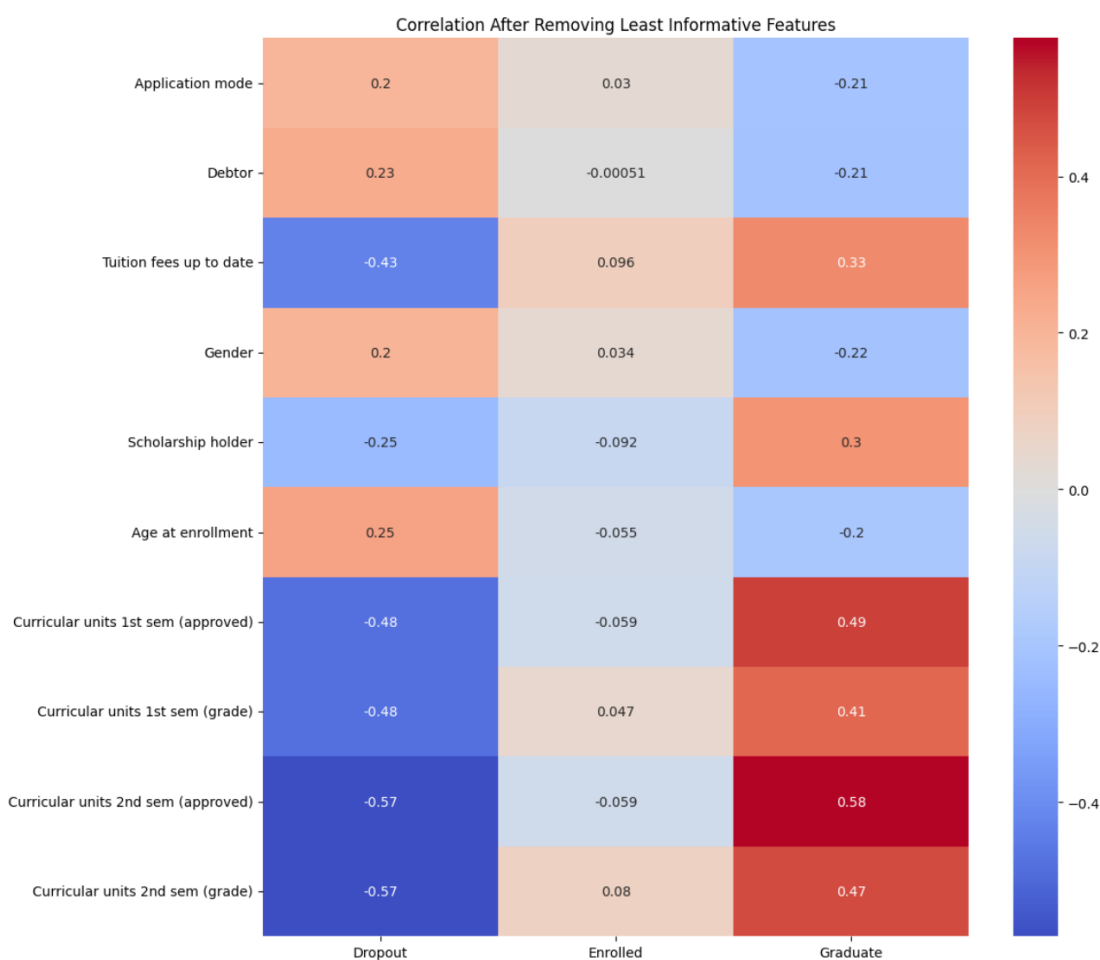
```
df["Target_Graduate"] = df["Target_Graduate"].astype(int)
```

+ Code + Markdown

پس از آن همبستگی ویژگی‌ها را با هر یک از این سه ویژگی جدید بررسی کردیم.



پس از آن در یک مرحله ۲۰ ویژگی با کمترین همبستگی را از دیتاست حذف کردیم و در مرحله‌ی دیگر روی دیتاست جدید ۶ ویژگی با کمترین همبستگی را نیز از دیتاست حذف کردیم تا به ۱۰ ویژگی با بیشترین همبستگی رسیدیم.



سپس داده‌ها با استفاده از نمونه‌گیری طبقه‌بندی شده به دو بخش آموزش (۸۰٪) و آزمون (۲۰٪) تقسیم شد.

۳-۱- فازی سازی ویژگی‌ها

برای فازی سازی ویژگی‌ها از تابع مثلثی زیر استفاده شد.

```
import numpy as np

def triangular_membership(x, a, b, c):
    return np.maximum(np.minimum((x - a) / (b - a), (c - x) / (c - b)), 0)
```

مقادیر انتخاب شده برای فازی سازی ویژگی‌های باینری و پیوسته به صورت زیر است.

```

# Application mode (1-57)
df_fuzzy["appmode_low"] = triangular_membership(df_fuzzy["Application mode"], 1, 10, 20)
df_fuzzy["appmode_med"] = triangular_membership(df_fuzzy["Application mode"], 15, 25, 35)
df_fuzzy["appmode_high"] = triangular_membership(df_fuzzy["Application mode"], 30, 45, 57)

# Age at enrollment (17-70)
df_fuzzy["age_low"] = triangular_membership(df_fuzzy["Age at enrollment"], 17, 20, 23)
df_fuzzy["age_med"] = triangular_membership(df_fuzzy["Age at enrollment"], 21, 26, 35)
df_fuzzy["age_high"] = triangular_membership(df_fuzzy["Age at enrollment"], 30, 40, 70)

# Curricular units 1st sem (approved) (0-26)
df_fuzzy["sem1_approved_low"] = triangular_membership(df_fuzzy["Curricular units 1st sem (approved)"], 0, 2, 4)
df_fuzzy["sem1_approved_med"] = triangular_membership(df_fuzzy["Curricular units 1st sem (approved)"], 3, 5, 7)
df_fuzzy["sem1_approved_high"] = triangular_membership(df_fuzzy["Curricular units 1st sem (approved)"], 6, 10, 26)

# Curricular units 1st sem (grade) (0-18.875)
df_fuzzy["sem1_grade_low"] = triangular_membership(df_fuzzy["Curricular units 1st sem (grade)"], 0, 5, 10)
df_fuzzy["sem1_grade_med"] = triangular_membership(df_fuzzy["Curricular units 1st sem (grade)"], 8, 12, 16)
df_fuzzy["sem1_grade_high"] = triangular_membership(df_fuzzy["Curricular units 1st sem (grade)"], 14, 17, 18.9)

# Curricular units 2nd sem (approved) (0-20)
df_fuzzy["sem2_approved_low"] = triangular_membership(df_fuzzy["Curricular units 2nd sem (approved)"], 0, 2, 4)
df_fuzzy["sem2_approved_med"] = triangular_membership(df_fuzzy["Curricular units 2nd sem (approved)"], 3, 5, 7)
df_fuzzy["sem2_approved_high"] = triangular_membership(df_fuzzy["Curricular units 2nd sem (approved)"], 6, 10, 20)

# Curricular units 2nd sem (grade) (0-18.57)
df_fuzzy["sem2_grade_low"] = triangular_membership(df_fuzzy["Curricular units 2nd sem (grade)"], 0, 5, 10)
df_fuzzy["sem2_grade_med"] = triangular_membership(df_fuzzy["Curricular units 2nd sem (grade)"], 8, 12, 16)
df_fuzzy["sem2_grade_high"] = triangular_membership(df_fuzzy["Curricular units 2nd sem (grade)"], 14, 17, 18.6)

# Debtor
df_fuzzy["debtor_no"] = 1 - df_fuzzy["Debtor"]
df_fuzzy["debtor_yes"] = df_fuzzy["Debtor"]

# Tuition fees up to date
df_fuzzy["fees_not_paid"] = 1 - df_fuzzy["Tuition fees up to date"]
df_fuzzy["fees_paid"] = df_fuzzy["Tuition fees up to date"]

# Gender (0: male, 1: female)
df_fuzzy["gender_male"] = 1 - df_fuzzy["Gender"]
df_fuzzy["gender_female"] = df_fuzzy["Gender"]

# Scholarship holder
df_fuzzy["no_scholarship"] = 1 - df_fuzzy["Scholarship holder"]
df_fuzzy["has_scholarship"] = df_fuzzy["Scholarship holder"]

```

بعد از آن ویژگی‌های مطلق در دیتاست حذف شد.

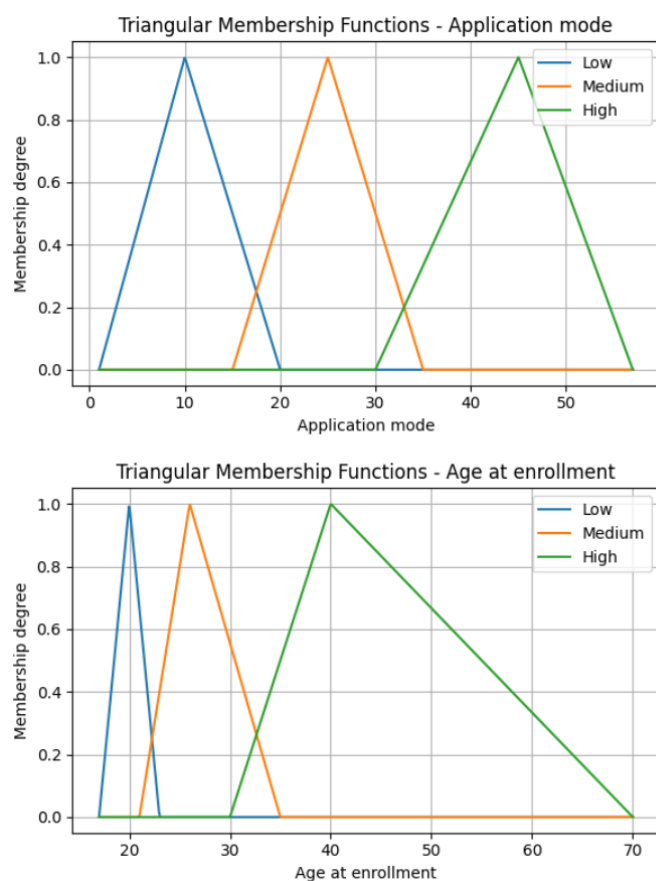
```

columns_to_drop = [
    "Application mode",
    "Debtor",
    "Tuition fees up to date",
    "Gender",
    "Scholarship holder",
    "Age at enrollment",
    "Curricular units 1st sem (approved)",
    "Curricular units 1st sem (grade)",
    "Curricular units 2nd sem (approved)",
    "Curricular units 2nd sem (grade)"
]

df_fuzzy.drop(columns=columns_to_drop, inplace=True)

```


در تصویر زیر نمودار توابع عضویت برای دو ویژگی Application mode و Age of enrollment آمده است.



۱-۴- استخراج قوانین فازی (روش Mendel-Wang)

برای تولید قوانین فازی از روش Mendel-Wang استفاده شد که کد مربوطه به این بخش در تصویر زیر قابل مشاهده است.

```
from sklearn.model_selection import train_test_split

y = df["Target"]

df_fuzzy_train, df_fuzzy_test, y_train, y_test = train_test_split(
    df_fuzzy, y, test_size=0.2, stratify=y, random_state=42
)
```

```

from collections import defaultdict
import pandas as pd
import numpy as np

# در یک دیتافریم y_train و df_fuzzy ترکیب
df_rules = df_fuzzy_train.copy()
df_rules['Target'] = y_train.values

rules = []

# تبدیل هر سطر به یک قانون فازی
for idx, row in df_rules.iterrows():
    antecedent = []
    mu_list = []

    for col in df_fuzzy.columns:
        val = row[col]
        if val > 0:
            antecedent.append(col)
            mu_list.append(val)

    # درجه عضویت کل برای این مقدمه
    mu = np.min(mu_list) # Minimum t-norm
    consequent = row['Target']

    rules.append((tuple(sorted(antecedent)), consequent, mu))

# ذخیره بهترین قانون برای هر مقدمه
best_rules = {}

for antecedent, consequent, weight in rules:
    key = (antecedent, consequent)
    if key not in best_rules or best_rules[key] < weight:
        best_rules[key] = weight

# حالا انتخاب بهترین قانون برای هر مقدمه یکتا
final_rules = {}

for (antecedent, consequent), weight in best_rules.items():
    if antecedent not in final_rules or final_rules[antecedent][1] < weight:
        final_rules[antecedent] = (consequent, weight)

# نهایی DataFrame تبدیل به
rules_df = pd.DataFrame([
    {'Antecedent': ' AND '.join(antecedent), 'Class': label, 'Weight': weight}
    for antecedent, (label, weight) in final_rules.items()
])

rules_df = rules_df.sort_values(by='Weight', ascending=False).reset_index(drop=True)

```

در تصویر زیر تعدادی از قوانین استخراج شده آمده که وزن آن‌ها برای انتخاب و برتری دادن آن‌ها نسبت به یکدیگر می‌باشد.

ترکیب (Crossover): روش cxTwoPoint برای ترکیب دو والد و تولید فرزندان جدید.
جهش (Mutation): تغییر تصادفی بیت‌ها با احتمال ۰.۵٪ با استفاده از (BitFlipMutation)
روند تکامل:
جمعیت اولیه به اندازه ۲۵ تولید می‌شود.
الگوریتم به مدت ۱۵ نسل اجرا می‌شود.
در پایان، بهترین فرد (کروموزوم) انتخاب شده و قوانین فازی مربوط به بیت‌های ۱ به عنوان قوانین منتخب
گزارش می‌شوند.

۱-۶- استنتاج فازی برای طبقه بندی

```
from collections import Counter

# مرحله استنتاج روی داده‌های تست
def fuzzy_inference(test_data, rules):
    predictions = []
    for i in range(len(test_data)):
        row = test_data.iloc[i]
        matched_classes = []

        for _, rule in rules.iterrows():
            antecedent = rule['Antecedent'].split(' AND ')
            match = all(row.get(feats, 0) == 1 for feats in antecedent if feats in row)
            if match:
                matched_classes.append(rule['Class'])

        # جمع: رأی‌گیری اکثریت
        if matched_classes:
            most_common = Counter(matched_classes).most_common(1)[0][0]
            predictions.append(most_common)
        else:
            predictions.append("Unknown")

    return predictions

# اجرای استنتاج روی df_fuzzy_test
final_preds = fuzzy_inference(df_fuzzy_test, selected_rules)

# نیستند Unknown فقط نمونه‌هایی را نگه می‌داریم که
filtered_preds = []
filtered_true = []
for pred, true in zip(final_preds, y_test):
    if pred != "Unknown":
        filtered_preds.append(pred)
        filtered_true.append(true)

# محاسبه دقت نهایی
if filtered_true:
    final_acc = accuracy_score(filtered_true, filtered_preds)
    print(f"🎯 دقت نهایی روی داده تست: {round(final_acc * 100, 2)}%")
    print(f"🔍 تعداد نمونه‌های طبقه‌بندی‌شده: {len(filtered_true)} از {len(y_test)}")
else:
    print("⚠️ هیچ نمونه‌ای در داده تست طبقه‌بندی نشد")
```

🎯 43.75 %دقت نهایی روی داده تست:

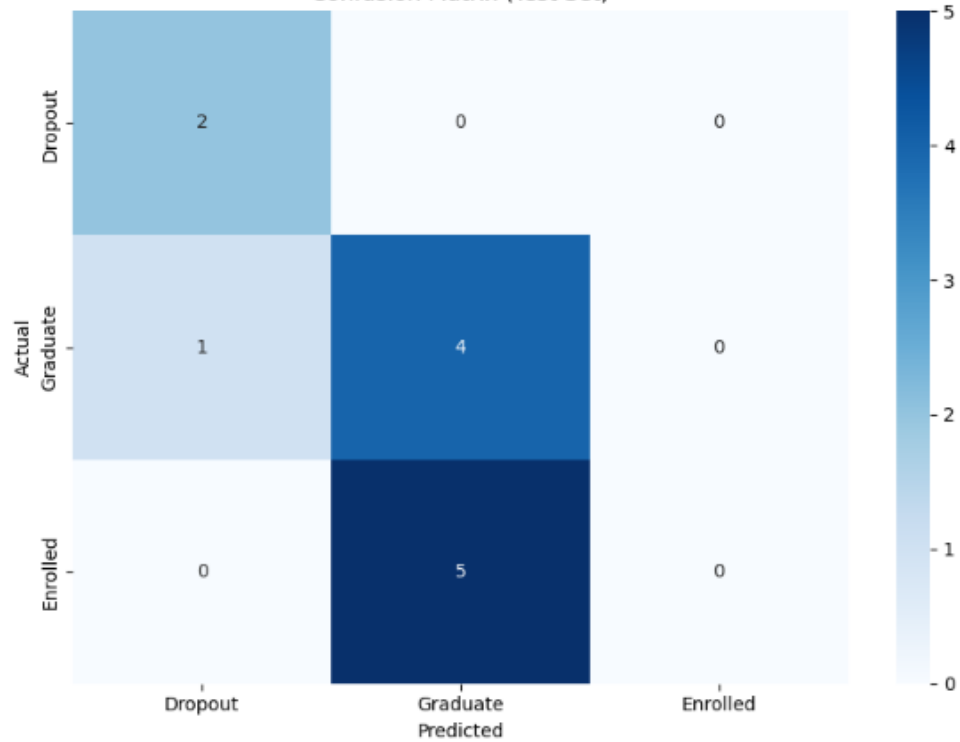
🔍 تعداد نمونه‌های طبقه‌بندی‌شده: 16 از 885

۱-۷- ارزیابی مدل

گزارش عملکرد مدل:

	precision	recall	f1-score	support
Dropout	0.67	1.00	0.80	2
Enrolled	0.00	0.00	0.00	5
Graduate	0.44	0.80	0.57	5
accuracy			0.50	12
macro avg	0.37	0.60	0.46	12
weighted avg	0.30	0.50	0.37	12

Confusion Matrix (Test Set)



- [1] "chatgpt," [Online]. Available: <https://chatgpt.com/>.
- [2] "youtube," [Online]. Available:
<https://www.youtube.com/watch?v=XuWPdFteIkQ>.
- [3] "youtube," [Online]. Available:
<https://www.youtube.com/watch?v=XACvF3TtywM>.
- [4] Kruse, Mostaghim, Borgelt, Braune and Steinbrecher, Computational intelligence: A methodological introduction, 2022.