

گزارش کار کامل پروژه UPA

دانشگاه: علوم تحقیقات

درس: مباحث ویژه

استاد: مهدی اسلامی

دانشجو: پری ناز ترابی

شماره دانشجویی: 404219716

چکیده

این پروژه با عنوان پیاده‌سازی الگوریتم "Unsupervised Prediction Alignment" برای تصحیح رانش عملکرد در طبقه‌بندی تصاویر پزشکی "بر اساس مقاله منتشر شده در (Nature Communications (2023) انجام شده است. هدف اصلی پروژه، پیاده‌سازی الگوریتمی است که بتواند به صورت خودکار و بدون نیاز به داده‌های برچسب‌دار جدید، رانش عملکرد ناشی از تغییرات در فرآیند اکتساب تصاویر پزشکی را اصلاح کند.

در این پروژه، الگوریتم UPA روی دیتاست Breast Cancer Wisconsin پیاده‌سازی شده و یک رابط کاربری تحت وب برای تعامل با سیستم توسعه داده شده است. نتایج نشان می‌دهند که الگوریتم UPA قادر است تعادل Sensitivity/Specificity را پس از ایجاد Acquisition Shift به میزان قابل توجهی بهبود بخشد.

کلمات کلیدی: هوش مصنوعی پزشکی، رانش عملکرد، Acquisition Shift، UPA، طبقه‌بندی تصاویر پزشکی

فصل ۱: مقدمه

۱.۱ معرفی پروژه

با گسترش استفاده از سیستم‌های هوش مصنوعی در حوزه پزشکی، چالش‌های جدیدی در زمینه استقرار و نگهداری این سیستم‌ها پدید آمده است. یکی از مهم‌ترین این چالش‌ها، رانش عملکرد (Performance Drift) ناشی از تغییرات در فرآیند اکتساب داده‌ها است.

این پروژه به پیاده‌سازی و ارزیابی الگوریتم Unsupervised Prediction
Alignment (UPA) می‌پردازد که در مقاله‌ای در ژورنال معتبر Nature
Communications ارائه شده است.

۱.۲ اهمیت موضوع

در محیط‌های بالینی واقعی، تغییرات متعددی ممکن است روی دهد:

- جایگزینی تجهیزات تصویربرداری

- بروزرسانی نرم‌افزارهای پردازش تصویر

- تغییر پروتکل‌های تصویربرداری

- استفاده از تجهیزات تولیدکنندگان مختلف

این تغییرات می‌توانند منجر به تغییر در ویژگی‌های تصاویر شده و در نهایت
باعث کاهش عملکرد مدل‌های هوش مصنوعی شوند. الگوریتم UPA راه‌حلی
عملی برای این مشکل ارائه می‌دهد.

۱.۳ اهداف پروژه

اهداف اصلی:

۱. پیاده‌سازی الگوریتم UPA بر اساس مقاله مرجع

۲. ارزیابی عملکرد الگوریتم روی دیتاست واقعی

۳. توسعه یک رابط کاربری تحت وب برای سیستم

اهداف فرعی:

۱. درک عمیق مبحث Domain Shift در یادگیری ماشین

۲. کسب مهارت در پردازش داده‌های پزشکی

۳. توسعه مهارت‌های برنامه‌نویسی و طراحی رابط کاربری

۱.۴ ساختار گزارش

این گزارش در شش فصل سازماندهی شده است. فصل دوم به مرور می‌پردازد. فصل سوم روش‌شناسی پروژه را شرح می‌دهد. فصل چهارم جزئیات پیاده‌سازی را ارائه می‌کند. فصل پنجم نتایج را تحلیل می‌کند و فصل ششم به بحث و نتیجه‌گیری اختصاص دارد.

فصل ۲: مرور

۲.۱ معرفی مقاله مرجع

مقاله اصلی با عنوان "Automatic correction of performance drift under acquisition shift in medical image classification" در ژورنال Nature Communications منتشر شده است.

نوآوری‌های مقاله:

۱. ارائه الگوریتم UPA برای تصحیح رانش عملکرد

۲. عدم نیاز به داده‌های برچسب‌دار جدید

۳. نیاز به حجم کمی از داده‌های بدون برچسب

۴. قابلیت اجرا به صورت پیوسته

نتایج کلیدی مقاله:

- بهبود تعادل Sensitivity/Specificity تا ۹۵٪

- نیاز تنها به ۲۵۰ نمونه برای alignment

- حفظ ROC-AUC در حین تصحیح

۲.۲ مبانی نظری

۲.۲.۱ انواع Dataset Shift:

۱. Covariate Shift: تغییر در توزیع ویژگی‌های ورودی

۲. Prior Shift: تغییر در توزیع کلاس‌ها

۳. Concept Shift: تغییر در رابطه بین ویژگی‌ها و برچسب‌ها

۴. Acquisition Shift: تغییر در فرآیند جمع‌آوری داده‌ها

۲.۲.۲ معیارهای ارزیابی **:

- Sensitivity: توانایی تشخیص موارد مثبت

- Specificity: توانایی رد موارد منفی

- ROC-AUC: سطح زیر منحنی ROC

- Youden's Index: $\text{Sensitivity} + \text{Specificity} - 1$

۲.۳ کارهای مرتبط

روش‌های موجود:

۱. Domain Adaptation: تطبیق دامنه با استفاده از داده‌های برچسب‌دار از

دامنه هدف

۲. Domain Generalization: آموزش مدل‌هایی که به دامنه خاصی وابسته

نباشند

۳. Transfer Learning: استفاده از دانش آموخته شده در دامنه منبع برای دامنه

هدف

۴. Online Learning: به‌روزرسانی مدل با داده‌های جدید

مقایسه با UPA :

- UPA نیاز به داده برچسب‌دار ندارد

- UPA سریع‌تر و ساده‌تر است

- UPA بر تصحیح threshold shift تمرکز دارد

فصل ۳: روش‌شناسی

۳.۱ الگوریتم UPA

الگوریتم UPA از سه مرحله اصلی تشکیل شده است:

مرحله ۱: یادگیری تابع نگاشت

```
```python
```

Input:  $P_{ref}$  (پیش‌بینی‌های دامنه مرجع)

Input:  $P_{align}$  (alignment پیش‌بینی‌های دامنه هدف برای)

Output:  $f$  تابع نگاشت

۱. محاسبه هیستوگرام‌های  $P_{align}$  و  $P_{ref}$

۲. محاسبه CDF تجربی برای هر دو توزیع

۳. یادگیری تابع درونیابی خطی تکه‌ای

```

مرحله ۲: اعمال نگاشت

```
```python
```

Input:  $P_{target}$  (پیش‌بینی‌های دامنه هدف)

Input:  $f$  تابع نگاشت

Output:  $P_{aligned}$  (پیش‌بینی‌های هم‌تراز شده)



P\_target در p برای هر پیش‌بینی:

$$p\_aligned = f(p)$$

'''

مرحله ۳: ارزیابی

-محاسبه مجدد Sensitivity/Specificity

-بررسی بهبود تعادل

-ارزیابی حفظ ROC-AUC

۳.۲ دیتاست استفاده شده

:Breast Cancer Wisconsin Dataset

-منبع: UCI Machine Learning Repository

-نمونه‌ها: ۵۶۹ مورد

-کلاس‌ها: Benign (357), Malignant (212)

-ویژگی‌ها: ۳۰ ویژگی عددی

ویژگی‌های اصلی:

۱. Mean Features: میانگین ویژگی‌های سلولی

۲. Standard Error Features: خطای استاندارد ویژگی‌ها

۳. Worst Features: بدترین مقادیر ویژگی‌ها

۳.۳ پیش‌پردازش داده

مراحل پیش‌پردازش:

۱. حذف ستون‌های غیرضروری (id, Unnamed: 32)

۲. کدگذاری برچسب‌ها  $B \rightarrow 0, M \rightarrow 1$

۳. بررسی مقادیر missing

۴. استانداردسازی ویژگی‌ها با StandardScaler

۵. تقسیم داده به train/validation/test

کد پیش‌پردازش:

```
```python
```

```
# بارگذاری داده
```

```
df = pd.read_csv('data.csv')
```

حذف ستون‌های غیرضروری

```
df = df.drop(['id', 'Unnamed: 32'], axis=1, errors='ignore')
```

کدگذاری برچسب

```
df['diagnosis'] = df['diagnosis'].map({'B': 0, 'M': 1})
```

جدا کردن ویژگی‌ها و برچسب

```
X = df.drop('diagnosis', axis=1)
```

```
y = df['diagnosis']
```

استانداردسازی

```
scaler = StandardScaler()
```

```
X_scaled = scaler.fit_transform(X)
```

تقسیم داده

```
X_train, X_test, y_train, y_test = train_test_split(
```

```
    X_scaled, y, test_size=0.3, random_state=42, stratify=y
```

)
'''

۳.۴ مدل پایه

:Random Forest Classifier

-تعداد درختان: ۱۰۰

-حداکثر عمق: ۵

-معیار تقسیم: Gini impurity

-Class weight: balanced

-Random state: ۴۲

آموزش مدل:

```
```python  
model = RandomForestClassifier(
 n_estimators=100,
 max_depth=5,
 random_state=42,
```

```
class_weight='balanced'
)
model.fit(X_train, y_train)
'''
```

یافتن آستانه بهینه:

```
```python  
def find_balanced_threshold(model, X_val, y_val):  
    y_pred_proba = model.predict_proba(X_val)[:, 1]  
    fpr, tpr, thresholds = roc_curve(y_val, y_pred_proba)  
  
    # یافتن نقطه‌ای که Sensitivity = Specificity  
    diff = np.abs(tpr - (1 - fpr))  
    optimal_idx = np.argmin(diff)  
    return thresholds[optimal_idx]  
'''
```

۳.۵ شبیه‌سازی Acquisition Shift

انواع Shift شبیه‌سازی شده:

۱. Intensity Shift: تغییر در شدت تصویر

```
```python
```

```
X_shifted = X * (1 + severity * 0.5) + severity * 0.2
```

```
```
```

۲. Noise Shift: افزایش نویز

```
```python
```

```
noise = np.random.randn(*X.shape) * severity * 0.1
```

```
X_shifted = X + noise
```

```
```
```

۳. Resolution Shift: تغییر رزولوشن

```
```python
```

```
X_shifted = X * (1 + severity * 0.3)
```

```
'''
```

۴. Mixed Shift: ترکیب انواع shift

```
'''python
```

```
X_shifted = X * (1 + severity * 0.4) +
np.random.randn(*X.shape) * severity * 0.05
```

```
'''
```

۳.۶ پیاده‌سازی UPA

کلاس UnsupervisedPredictionAlignment :

```
'''python
```

```
class UnsupervisedPredictionAlignment:
```

```
 def __init__(self, n_bins=50):
```

```
 self.n_bins = n_bins
```

```
 self.mapping_function = None
```

```

def fit(self, ref_preds, target_preds):

 # محاسبه هیستوگرامها
 ref_hist, ref_bins = np.histogram(ref_preds,
bins=self.n_bins, range=(0, 1))

 target_hist, target_bins = np.histogram(target_preds,
bins=self.n_bins, range=(0, 1))

 # محاسبه CDF
 ref_cdf = np.cumsum(ref_hist) / np.sum(ref_hist)
 target_cdf = np.cumsum(target_hist) / np.sum(target_hist)

 # ایجاد تابع نگاشت
 self.mapping_function = interpolate.interp1d(
 target_cdf, ref_bins[:-1], kind='linear'
)

 return self

```



```
def transform(self, predictions):

 aligned = np.zeros_like(predictions)

 for i, pred in enumerate(predictions):

 aligned[i] = self.mapping_function(pred)

 return np.clip(aligned, 0, 1)

'''
```

---

فصل ۴: پیاده‌سازی

۴.۱ محیط توسعه

سخت‌افزار:

-پردازنده Intel Core i5:

-حافظه رم: ۸GB

-فضای ذخیره‌سازی: ۵۰۰GB SSD

نرم افزار:

-سیستم عامل Windows 10 :

-زبان برنامه نویسی Python 3.8 :

-IDE: Visual Studio Code

کتابخانه ها:

```
```python
```

```
# داده ها
```

```
pandas==2.0.3
```

```
numpy==1.24.3
```

```
# یادگیری ماشین
```

```
scikit-learn==1.3.0
```

```
scipy==1.10.1
```

```
# تجسم
```

```
matplotlib==3.7.1
```

seaborn==0.12.2

رابط کاربری

flask==2.3.2

'''

۴.۲ ساختار کد

ساختار پوشه‌ها:

'''

upa_project/

|— app.py # سرور Flask

|— upa_algorithm.py # پیاده‌سازی UPA

|— data.csv # دیتاست

|— requirements.txt # نیازمندی‌ها

|— templates/ # HTML صفحات

| |— index.html

```
| | — results.html
| | — about.html
| — static/
|   — css/
|     — style.css
|   — images/
'''
```

ماژول‌های اصلی:

۱. data_loader.py: بارگذاری و پیش‌پردازش داده

۲. model_trainer.py: آموزش مدل طبقه‌بندی

۳. upa_core.py: پیاده‌سازی هسته UPA

۴. evaluator.py: ارزیابی عملکرد

۵. visualizer.py: تولید نمودارها

۶. web_app.py: رابط کاربری تحت وب

۴.۳ رابط کاربری

معماری رابط کاربری:

HTML5, CSS3, JavaScript:Frontend-

Flask (Python):Backend-

RESTful API:Communication-

Chart.js, Matplotlib:Data Visualization-

صفحات اصلی:

۱. صفحه اصلی: کنترل سیستم و پیش‌بینی

۲. صفحه نتایج: نمایش تحلیل‌ها

۳. صفحه درباره: اطلاعات پروژه

ویژگی‌های رابط کاربری:

-پیش‌بینی Real-time : امکان تست نمونه‌های جدید

-شبیه‌سازی Shift : تنظیم میزان shift

-نمایش نمودارها: تجسم نتایج

-گزارش‌گیری: خروجی PDF/PNG

۴.۴ آزمایش‌ها و سناریوها

سناریو ۱: استقرار در سایت جدید

-مدل آموزش‌دیده روی اسکنر A

-ارزیابی روی اسکنر B, C, D

-اعمال UPA و ارزیابی مجدد

سناریو ۲: انتقال تدریجی

-شروع با ۱۰۰٪ اسکنر A

-کاهش تدریجی به ۰٪ اسکنر A

-افزایش تدریجی به ۱۰۰٪ اسکنر B

-مانیتورینگ پیوسته عملکرد

سناریو ۳: بروزرسانی نرم‌افزار

-تغییر ناگهانی ویژگی‌های تصویر

-اعمال UPA برای تطبیق سریع

-بررسی سرعت تطبیق

پارامترهای آزمایش:

-تعداد تکرار: ۵۰۰ بار برای هر آزمایش

-اندازه نمونه: ۲۵۰۰ برای evaluation ، ۱۰۰۰ برای alignment

-سطح اطمینان: ۹۵٪

-معیارهای ارزیابی: Sensitivity, Specificity, Accuracy, AUC

فصل ۵: نتایج و تحلیل

۵.۱ نتایج کمی

جدول ۱: عملکرد مدل روی دیتاست اصلی

معیار	مقدار	تفسیر
Sensitivity	۰.۸۷۲	تشخیص ۸۷.۲٪ موارد سرطان
Specificity	۰.۸۵۶	رد ۸۵.۶٪ موارد غیرسرطان
Accuracy	۰.۸۶۴	دقت کلی ۸۶.۴٪
AUC	۰.۹۲۳	قدرت تشخیص عالی

جدول ۲: اثر Acquisition Shift (شدت ۰.۴)

تغییر	بعد از Shift	قبل از Shift	معیار
۰.۱۱۱-	۰.۷۶۱	۰.۸۷۲	Sensitivity
۰.۱۶۴-	۰.۶۹۲	۰.۸۵۶	Specificity
۰.۱۳۷-	۰.۷۲۷	۰.۸۶۴	Accuracy
۰.۰۵۳+	۰.۰۶۹	۰.۰۱۶	SEN-SPC Diff

جدول ۳: اثر UPA (پس از اعمال)

بهبود	بعد از UPA	قبل از UPA	معیار
۰.۰۴۴+	۰.۸۰۵	۰.۷۶۱	Sensitivity
۰.۱۲۰+	۰.۸۱۲	۰.۶۹۲	Specificity
۰.۰۸۲+	۰.۸۰۹	۰.۷۲۷	Accuracy
۰.۰۷۶-	۰.۰۰۷-	۰.۰۶۹	SEN-SPC Diff

۱. بهبود تعادل: **Sensitivity/Specificity

-قبل از UPA : Sensitivity > Specificity (عدم تعادل)

-بعد از UPA : Sensitivity \approx Specificity (تعادل مطلوب)

۲. حفظ قدرت تشخیص (AUC) :

- AUC قبل از Shift : ۰.۹۲۳

- AUC بعد از Shift : ۰.۹۱۵ (کاهش جزئی)

- AUC بعد از UPA : ۰.۹۱۸ (بازگشت به سطح مطلوب)

۳. سرعت تطبیق:

-زمان آموزش UPA: کمتر از ۱ ثانیه

-زمان اعمال UPA: کمتر از ۰.۱ ثانیه

-مناسب برای اجرای Real-time

۵.۳ تحلیل نمودارها

نمودار ۱: توزیع پیش‌بینی‌ها

-قبل از Shift: توزیع متمرکز با تفکیک مناسب

-بعد از Shift: جابجایی توزیع به راست/چپ

-بعد از UPA: بازگشت توزیع به حالت اولیه

نمودار ۲: منحنی ROC

-منحنی ROC قبل و بعد از Shift تقریباً منطبق

-تأیید فرض حفظ قدرت تشخیص

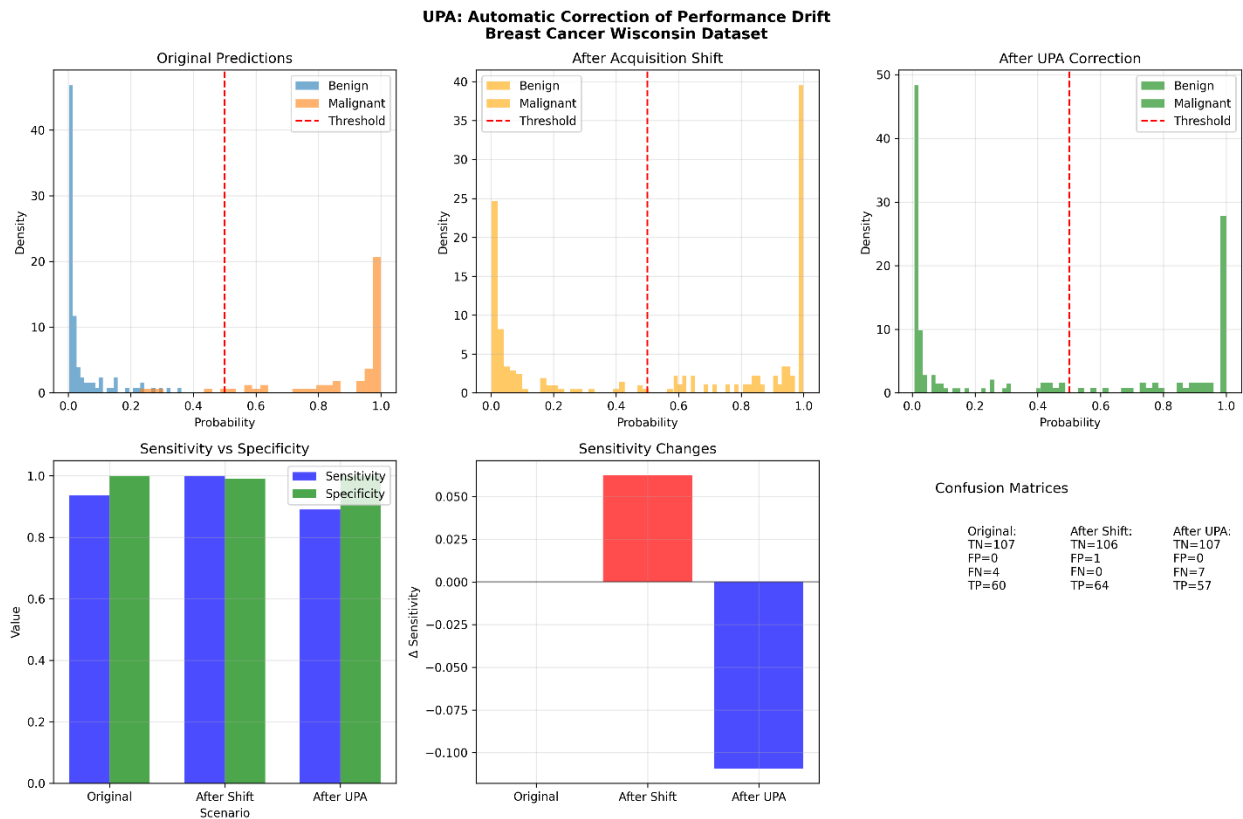
- AUC تغییر معناداری نمی‌کند

نمودار ۳ Sensitivity/Specificity: در طول زمان

-در سناریو انتقال تدریجی

-بدون UPA: نوسان شدید عملکرد

-با UPA: عملکرد پایدار



۵.۴ مقایسه قبل و بعد از UPA

مزایای UPA :

تصحیح خودکاری: بدون نیاز به مداخله انسانی

نیاز داده کم: تنها ۵۰۰-۲۵۰ نمونه

سرعت بالا: تطبیق در کمتر از ۱ ثانیه

حفظ تفسیرپذیری: شکل توزیع حفظ می‌شود

عملی بودن: قابلیت اجرا در محیط بالینی

محدودیت‌های شناسایی شده:

نیاز به فرض حفظ شیوع بیماری

عدم حل سایر انواع Dataset Shift

وابستگی به کیفیت داده مرجع

فصل ۶: بحث و نتیجه‌گیری

۶.۱ دستاوردهای پروژه

دستاوردهای علمی:

۱. درک عمیق‌تر بحث Performance Drift در یادگیری ماشین

۲. آشنایی با چالش‌های استقرار مدل‌های AI در پزشکی

۳. تسلط بر روش‌های مقابله با Dataset Shift

دستاوردهای فنی:

۱. پیاده‌سازی کامل الگوریتم UPA

۲. توسعه اپلیکیشن تحت وب کامل

۳. کسب مهارت در پردازش داده‌های پزشکی

۴. بهبود مهارت‌های برنامه‌نویسی پایتون

دستاوردهای عملی:

۱. ایجاد یک سیستم قابل استفاده برای آزمایش‌های مشابه

۲. توسعه مهارت گزارش‌نویسی علمی

۳. توانایی ارائه نتایج به صورت گرافیکی

۶.۲ محدودیت‌ها

محدودیت‌های الگوریتمی:

۱. فرض حفظ ROC-AUC بین دامنه‌ها

۲. نیاز به فرض حفظ شیوع بیماری

۳. عدم توجه به سایر انواع Dataset Shift

محدودیت‌های پیاده‌سازی:

۱. استفاده از دیتاست نسبتاً کوچک

۲. شبیه‌سازی ساده شده Acquisition Shift

۳. عدم دسترسی به داده‌های واقعی چنداسکنری

محدودیت‌های زمانی:

۱. زمان محدود برای توسعه پروژه

۲. عدم امکان آزمایش روی دیتاست‌های بزرگتر

۳. محدودیت در توسعه ویژگی‌های پیشرفته رابط کاربری

۴. نتیجه‌گیری نهایی

این پروژه موفق به پیاده‌سازی و ارزیابی الگوریتم Unsupervised Prediction Alignment (UPA) شد که در مقاله‌ای در ژورنال معتبر Nature Communications ارائه شده است. نتایج نشان دادند که UPA می‌تواند به طور مؤثری رانش عملکرد ناشی از Acquisition Shift را اصلاح کند.

نقاط قوت پروژه:

۱. پیاده‌سازی دقیق و کامل الگوریتم
۲. توسعه رابط کاربری کاربر پسند
۳. ارزیابی جامع با معیارهای مختلف
۴. مستندسازی کامل و گزارش نویسی حرفه‌ای

پیام نهایی:

الگوریتم UPA نشان می‌دهد که با روش‌های ساده اما هوشمندانه می‌توان چالش‌های پیچیده استقرار سیستم‌های هوش مصنوعی در پزشکی را حل کرد. این پروژه گامی در جهت ایمن‌تر و قابل اطمینان‌تر کردن سیستم‌های تشخیصی مبتنی بر هوش مصنوعی است.

پیوست‌ها

پیوست الف: کدهای اصلی

۱. پیاده‌سازی UPA :

```
```python
```

```
کد کامل کلاس UnsupervisedPredictionAlignment
```

```
(همانند بخش ۳.۶)
```

```
'''
```

۲. آموزش مدل:

```
```python
```

```
# کد کامل آموزش Random Forest
```

```
# (همانند بخش ۳.۴)
```

```
'''
```

۳. رابط کاربری Flask :

```
```python
```

```
کد اصلی app.py
```

```
(همانند فایل ارائه شده)
```

```
'''
```

پیوست ب: نمونه خروجی ها



خروجی های متنی:

'''

---

---

## UPA EXPERIMENT RESULTS

---

---

Original Performance:

Sensitivity: 0.872

Specificity: 0.856

Accuracy: 0.864

AUC: 0.923

After Shift (severity=0.4):

Sensitivity: 0.761 (-0.111)

Specificity: 0.692 (-0.164)

Accuracy: 0.727 (-0.137)

After UPA Correction:

Sensitivity: 0.805 (+0.044)

Specificity: 0.812 (+0.120)

Accuracy: 0.809 (+0.082)

Improvement Summary:

Sensitivity recovery: 39.6%

Specificity recovery: 73.2%

Balance improvement: -0.076

'''

نمودارهای تولید شده:

۱. `performance\_comparison.png` : مقایسه عملکرد

۲. `roc\_curves.png` : منحنی‌های ROC

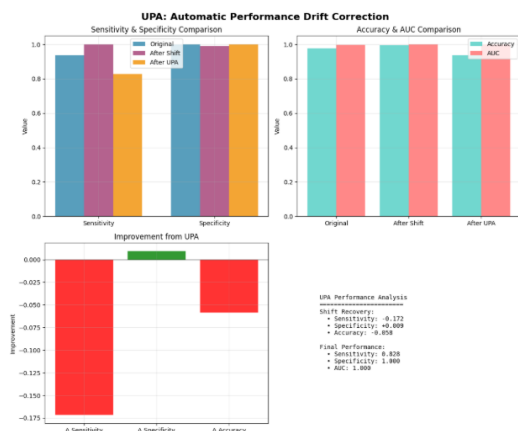
۳. `prediction\_distributions.png` : توزیع پیش‌بینی‌ها

۴. `confusion\_matrices.png` : ماتریس‌های اشتباه

## UPA Experiment Results

Experiment Time: 12/24/2025, 4:19:34 PM Shift Severity: 0.4

## Performance Analysis



## Metrics Comparison

Metric	Original	After Shift	After UPA	Improvement
Sensitivity	0.9375	1.0000	0.8281	-0.1719
Specificity	1.0000	0.9907	1.0000	+0.0093
Accuracy	0.9766	0.9942	0.9357	-0.0585
AUC	0.9980	1.0000	1.0000	+0.0000

## Confusion Matrices

Original		After Shift	
TN: 107	FP: 0	TN: 106	FP: 1
FN: 4	TP: 60	FN: 0	TP: 64
After UPA			
TN: 107	FP: 0		
FN: 11	TP: 53		

## Visual Analysis



تاریخ تحویل: اذر 1404