

심혈관 질환 예측 모델

전이학습을 활용하여

...

의료 프로젝트 팀: 고나연, 오승현, 유소은, 장서진

2025-08-24 아웃라이어 최종 발표회



목차 Contents



...

01 고나연
10년 후 심혈관계 질병 발병
위험도 예측 및 조언 웹앱

03 유소은
키워드 1 / 키워드 2 / 키워드 3

...

02 오승현
심혈관 질환 예측을 위한 웹앱 개발

04 장서진
키워드 1 / 키워드 2 / 키워드 3

02



심혈관 질환 예측을 위한 웹앱 개발

전이학습과 streamlit 사용 시각화

“ 프로젝트 개요

•
•
•

제목: 심혈관 질환의 위험도 예측 웹 애플리케이션 개발

목표: 심혈관 질환의 위험 예측 모델을 개발하고, streamlit
을 이용한 웹앱으로 배포하여 사용자 친화적인 예측 및 개
선 가이드를 제공

사용자의 건강 정보를 입력하면, 심혈관 질환 발병 확률과
예방법을 알수 있다. 추가적으로, 혈압 조절시 개선된 발
병 확률을 시각적으로 보여준다.

”



데이터 전처리



```
df['age'] = (df['age'] / 365).astype(int)
```

- `df = df[(df['ap_hi'] > 50) & (df['ap_hi'] < 250)]`
- `df = df[(df['ap_lo'] > 30) & (df['ap_lo'] < 200)]`
- `df = df[(df['height'] > 100) & (df['height'] < 250)]`
- `df = df[(df['weight'] > 30) & (df['weight'] < 250)]`

```
df['BMI'] = df['weight'] / ( (df['height'] / 100) ** 2 )
```

	id	age	gender	height	weight	ap_hi	ap_lo	cholesterol	gluc	smoke	alco	active	cardio
0	0	18393	2	168	62.0	110	80	1	1	0	0	1	0
1	1	20228	1	156	85.0	140	90	3	1	0	0	1	1
2	2	18857	1	165	64.0	130	70	3	1	0	0	0	1
3	3	17623	2	169	82.0	150	100	1	1	0	0	1	1
4	4	17474	1	156	56.0	100	60	1	1	0	0	0	0



	id	age	gender	height	weight	ap_hi	ap_lo	cholesterol	gluc	smoke	alco	active	cardio	BMI
0	0	50	2	168	62.0	110	80	1	1	0	0	1	0	21.967120
1	1	55	1	156	85.0	140	90	3	1	0	0	1	1	34.927679
2	2	51	1	165	64.0	130	70	3	1	0	0	0	1	23.507805
3	3	48	2	169	82.0	150	100	1	1	0	0	1	1	28.710479
4	4	47	1	156	56.0	100	60	1	1	0	0	0	0	23.011177



Optimization terminated successfully.
Current function value: 0.561039
Iterations 6

Logit Regression Results

```
=====
Dep. Variable:      cardio  No. Observations:      68736
Model:              Logit   Df Residuals:          68723
Method:             MLE     Df Model:              12
Date:              Thu, 31 Jul 2025  Pseudo R-squ.:      0.1905
Time:              13:11:23  Log-Likelihood:    -38564.
converged:          True     LL-Null:          -47641.
Covariance Type:    nonrobust  LLR p-value:       0.000
=====
```

	coef	std err	z	P> z	[0.025	0.975]
const	-10.1161	0.765	-13.221	0.000	-11.616	-8.616
age	0.0514	0.001	38.045	0.000	0.049	0.054
gender	-0.0123	0.022	-0.558	0.577	-0.056	0.031
height	-0.0109	0.005	-2.341	0.019	-0.020	-0.002
weight	0.0181	0.005	3.759	0.000	0.009	0.028
ap_hi	0.0531	0.001	60.969	0.000	0.051	0.055
ap_lo	0.0166	0.001	12.252	0.000	0.014	0.019
cholesterol	0.4986	0.016	31.957	0.000	0.468	0.529
gluc	-0.1193	0.018	-6.748	0.000	-0.154	-0.085
smoke	-0.1439	0.035	-4.136	0.000	-0.212	-0.076
alco	-0.2093	0.042	-4.954	0.000	-0.292	-0.127
active	-0.2281	0.022	-10.420	0.000	-0.271	-0.185
BMI	-0.0189	0.013	-1.492	0.136	-0.044	0.006

로지스틱 회귀

성능평가



	precision	recall	f1-score	support
0	0.70	0.77	0.74	6891
1	0.75	0.67	0.71	6857
accuracy			0.72	13748
macro avg	0.73	0.72	0.72	13748
weighted avg	0.73	0.72	0.72	13748

-
-
-

[Random Forest] 성능 평가

	precision	recall	f1-score	support
0	0.69	0.69	0.69	6891
1	0.69	0.69	0.69	6857
accuracy			0.69	13748
macro avg	0.69	0.69	0.69	13748
weighted avg	0.69	0.69	0.69	13748

• • •

Random Forest

```
model_rf = RandomForestClassifier(random_state=42)
model_rf.fit(X_train, y_train)
y_pred_rf = model_rf.predict(X_test)
print("\n[Random Forest] 성능 평가")
print(classification_report(y_test, y_pred_rf))
```



[LightGBM] 성능 평가

	precision	recall	f1-score	support
0	0.72	0.77	0.74	6891
1	0.75	0.70	0.72	6857
accuracy			0.73	13748
macro avg	0.73	0.73	0.73	13748
weighted avg	0.73	0.73	0.73	13748

• • •

LightGBM

```
model_lgb = LGBMClassifier(random_state=42)
model_lgb.fit(X_train, y_train)
y_pred_lgb = model_lgb.predict(X_test)
print("\n[LightGBM] 성능 평가")
print(classification_report(y_test, y_pred_lgb))
```

[XGBoost] 성능 평가				
	precision	recall	f1-score	support
0	0.72	0.76	0.74	6891
1	0.75	0.69	0.72	6857
accuracy			0.73	13748
macro avg	0.73	0.73	0.73	13748
weighted avg	0.73	0.73	0.73	13748

• • •

XGBoost



```
model_xgb = XGBClassifier(use_label_encoder=False,
eval_metric='logloss', random_state=42)
model_xgb.fit(X_train, y_train)
y_pred_xgb = model_xgb.predict(X_test)
print("\n[XGBoost] 성능 평가")
print(classification_report(y_test, y_pred_xgb))
```

[Gradient Boosting] 성능 평가				
	precision	recall	f1-score	support
0	0.72	0.77	0.74	6891
1	0.75	0.70	0.72	6857
accuracy			0.73	13748
macro avg	0.73	0.73	0.73	13748
weighted avg	0.73	0.73	0.73	13748

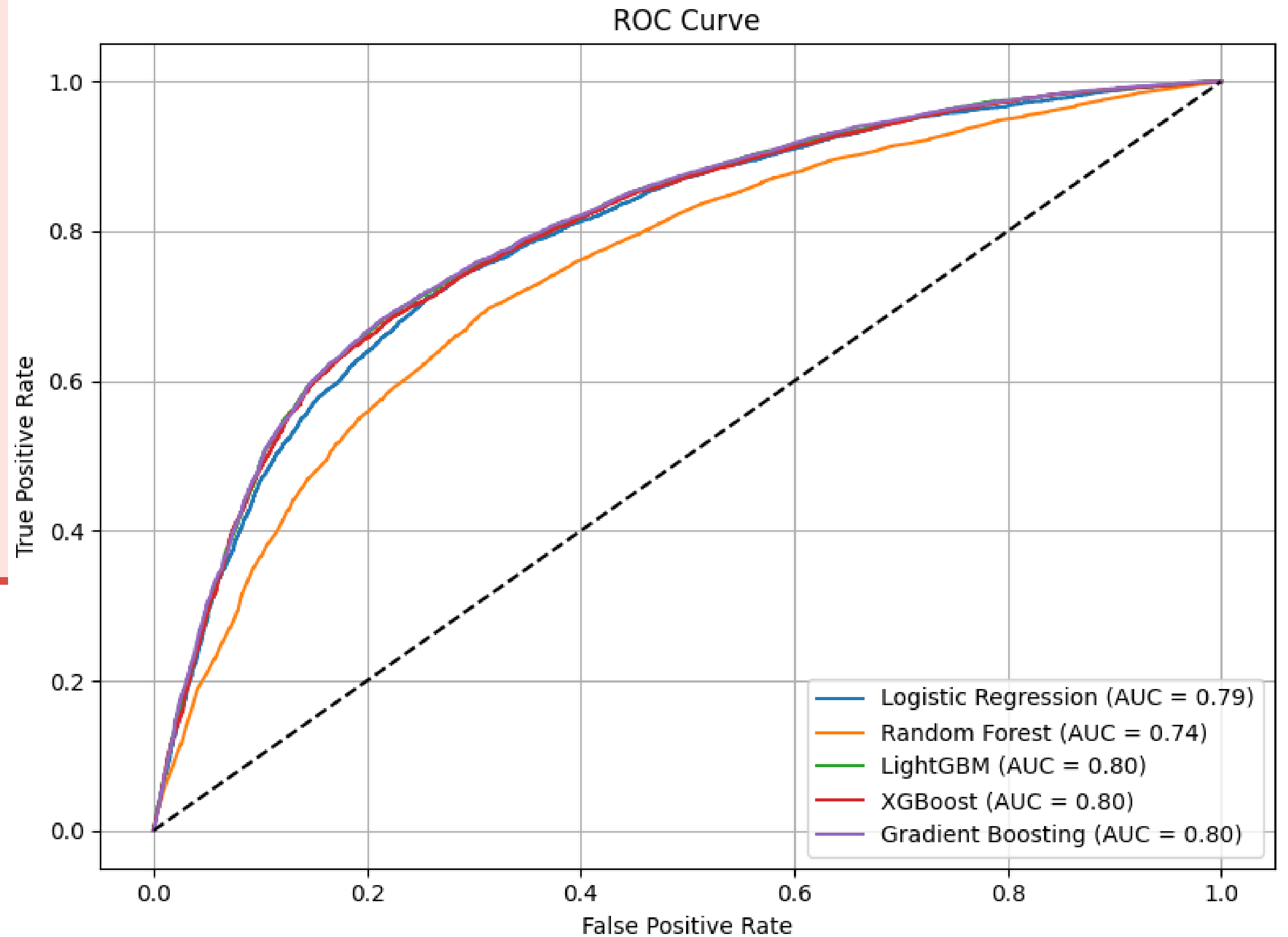
• • •

Gradient Boosting

```
model_gb = GradientBoostingClassifier(random_state=42)
model_gb.fit(X_train, y_train)
y_pred_gb = model_gb.predict(X_test)
print("\n[Gradient Boosting] 성능 평가")
print(classification_report(y_test, y_pred_gb))
```


ROC Curve

-
-
-



전이학습

```
heart = pd.read_csv('/content/drive/MyDrive/heart.csv')
heart_mapped = heart[['age', 'sex', 'chol', 'trestbps', 'fbs', 'target']].copy()
heart_mapped.columns = ['age', 'gender', 'cholesterol', 'ap_hi', 'gluc', 'cardio']

X_heart = heart_mapped.drop(columns=['cardio']).values
y_heart = heart_mapped['cardio'].values
Xh_train, Xh_val, yh_train, yh_val = train_test_split(X_heart, y_heart, test_size=0.2,
random_state=42)

pretrained_model = TabNetClassifier(seed=42)
pretrained_model.fit(
    X_train=Xh_train, y_train=yh_train,
    eval_set=[(Xh_val, yh_val)],
    eval_metric=['accuracy'],
    max_epochs=100,
    patience=10,
    batch_size=256,
    virtual_batch_size=128
)
pretrained_model.save_model("/content/tabnet_heart_pretrained")
```



전이학습

```
cardio = pd.read_csv('/content/drive/MyDrive/cardio_train.csv', sep=';')  
cardio_mapped = cardio[['age', 'gender', 'cholesterol', 'ap_hi', 'gluc', 'cardio']].dropna()
```

```
X_cardio = cardio_mapped.drop(columns=['cardio']).values  
y_cardio = cardio_mapped['cardio'].values  
Xc_train, Xc_val, yc_train, yc_val = train_test_split(X_cardio, y_cardio, test_size=0.2,  
random_state=42)
```

```
finetune_model = TabNetClassifier()  
finetune_model.load_model("/content/tabnet_heart_pretrained.zip")
```

```
finetune_model.fit(  
    X_train=Xc_train, y_train=yc_train,  
    eval_set=[(Xc_val, yc_val)],  
    eval_metric=['accuracy'],  
    max_epochs=50,  
    patience=5,  
    batch_size=256,  
    virtual_batch_size=128  
)
```

```
# 6. 성능 평가  
preds = finetune_model.predict(Xc_val)  
accuracy = accuracy_score(yc_val, preds)  
print(f"전이학습 이후 정확도: {accuracy:.4f}")
```

→ 전이학습 이후 정확도: 0.7031



Streamlit 시각화

XGBoost 사용

<https://cardio-project-q2btpa92nykgwv7ooabl6k.streamlit.app/>

건강 정보 입력

나이

38

성별

☐ 남성

☒ 여성

키 (cm)

162

몸무게 (kg)

65

수축기 혈압

120

이완기 혈압

53

콜레스테롤 등급

1

혈당 등급

1

☐ 흡연 여부

☒ 음주 여부

☐ 운동을 규칙적으로 하나요?

❤️ 당신의 심혈관 건강은 안전한가요?

XGBoost 기반 심혈관 질환 위험 예측 & 건강 개선 제안

사용자의 건강 정보 입력 시 위험도를 예측하고, 주요 개선 항목도 제시합니다.

📊 예측 결과

심혈관 질환 위험도

28.56%

개선 제안:

💡 음주 줄이기

💡 운동 시작하기

시뮬레이터: 혈압 조정 시 위험도 변화

수축기 혈압 (mmHg)

123

이완기 혈압 (mmHg)

57

혈압을 120/53 → 123/57 mmHg로 조정하면

위험도는 24.60%입니다.

Thank you

발표를 들어주셔서

감사합니다

<https://github.com/Launa-0/cardio-project>

...

