

04.

Transfer Learning



전이학습

- 이미 훈련된 모델을 다른 작업에 이용하는 것
- DeepFace: 사진의 얼굴을 자동 인식해 벡터로 변환해 주는 모델
 - ① Selenium 이용, 22.04.22 ~ 24.04.16 간 이창용 총재 뉴스기사 사진 크롤링 (731장)
 - ② DeepFace 이용, 각 사진을 벡터로 변환하여 데이터셋 행렬 구성
 - ③ Tensorflow 이용, 촬영된 시기의 금리 인상/동결 여부를 예측하는 새로운 모델 훈련
- 데이터 출처: 뉴시스 331건, 뉴스원 324건, 연합뉴스 76건

04.

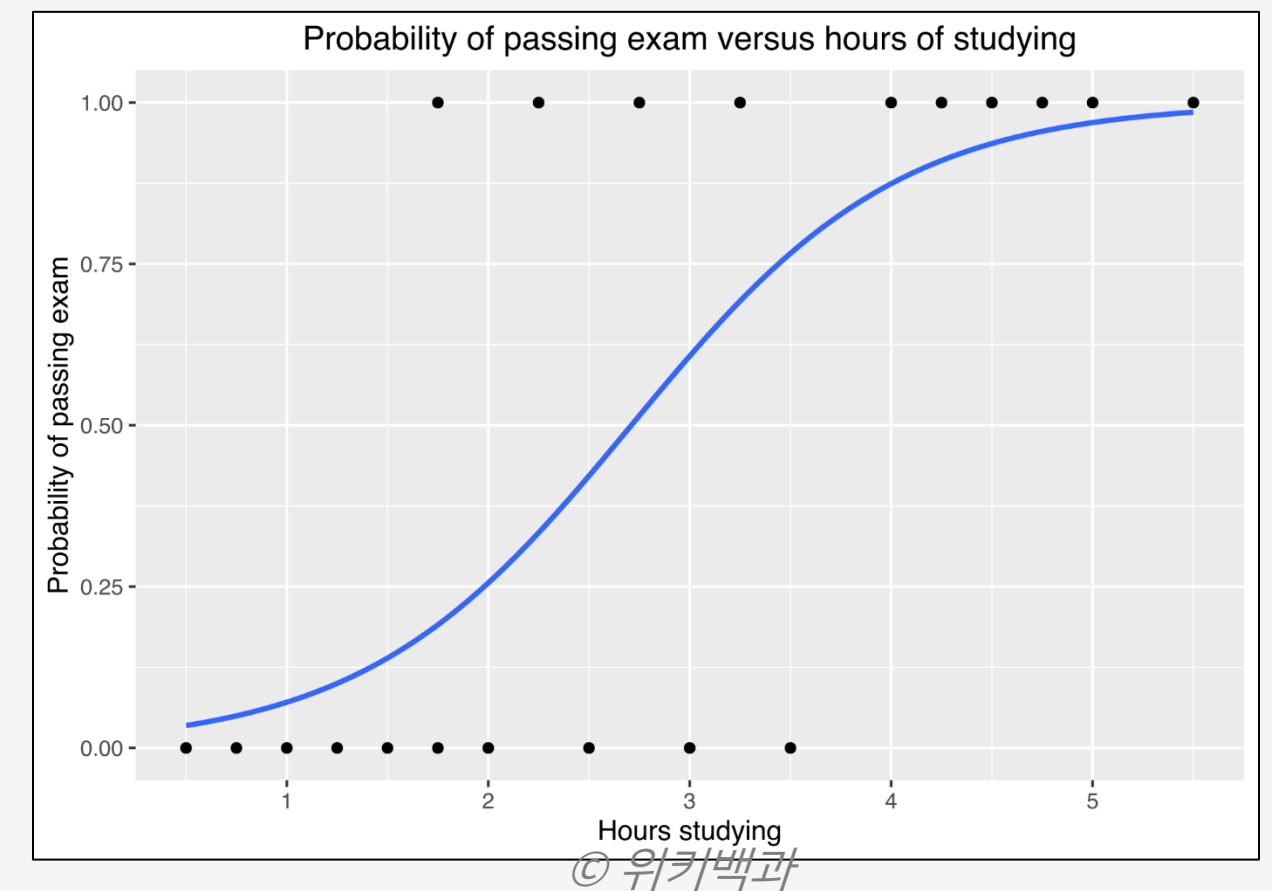
Transfer Learning

1. Logistic Regression

- 제일 간단한 이진분류 모형
- X: DeepFace가 사전에 분석한, 각 사진의 감성분석 벡터값(7차원)을 모은 행렬
- y: 금리 인상 시 1(234장), 동결 시 0(497장)
- 511장은 train, 220장은 test 데이터로 배정
 - y의 0과 1 비율은 train, test 양쪽 동일
- 결과: test 정확도 약 68.18%

	angry	disgust	fear	happy	sad	surprise	neutral
0	6.002435	6.477609e-07	16.222183	1.491965e-02	10.101155	6.668258e+01	9.767249e-01
1	0.000010	1.664180e-15	87.325972	1.117121e-05	0.000042	1.267397e+01	1.474510e-09
2	0.000005	2.633513e-16	0.795042	2.231258e-10	0.000060	9.920489e+01	1.805616e-09
3	0.000071	8.926598e-12	12.364900	1.365275e-04	87.633210	9.412999e-09	1.682087e-03
4	0.008719	2.687708e-11	9.325602	6.307153e-04	0.000098	9.066495e+01	1.343985e-06
...
726	0.216535	1.978689e-04	95.150954	3.226303e-02	4.007768	1.021479e-01	4.901287e-01
727	0.069519	1.202964e-07	43.616325	2.775884e-01	3.099770	5.201704e+01	9.197577e-01
728	0.016195	1.134596e-10	7.468444	1.389880e-03	79.998188	9.335746e+00	3.180039e+00
729	0.022460	1.240496e-06	0.552143	3.722562e-04	99.371433	6.114994e-09	5.359298e-02
730	0.035194	1.210611e-10	25.264496	4.079604e-02	0.001168	7.465835e+01	3.486468e-07

731 rows × 7 columns

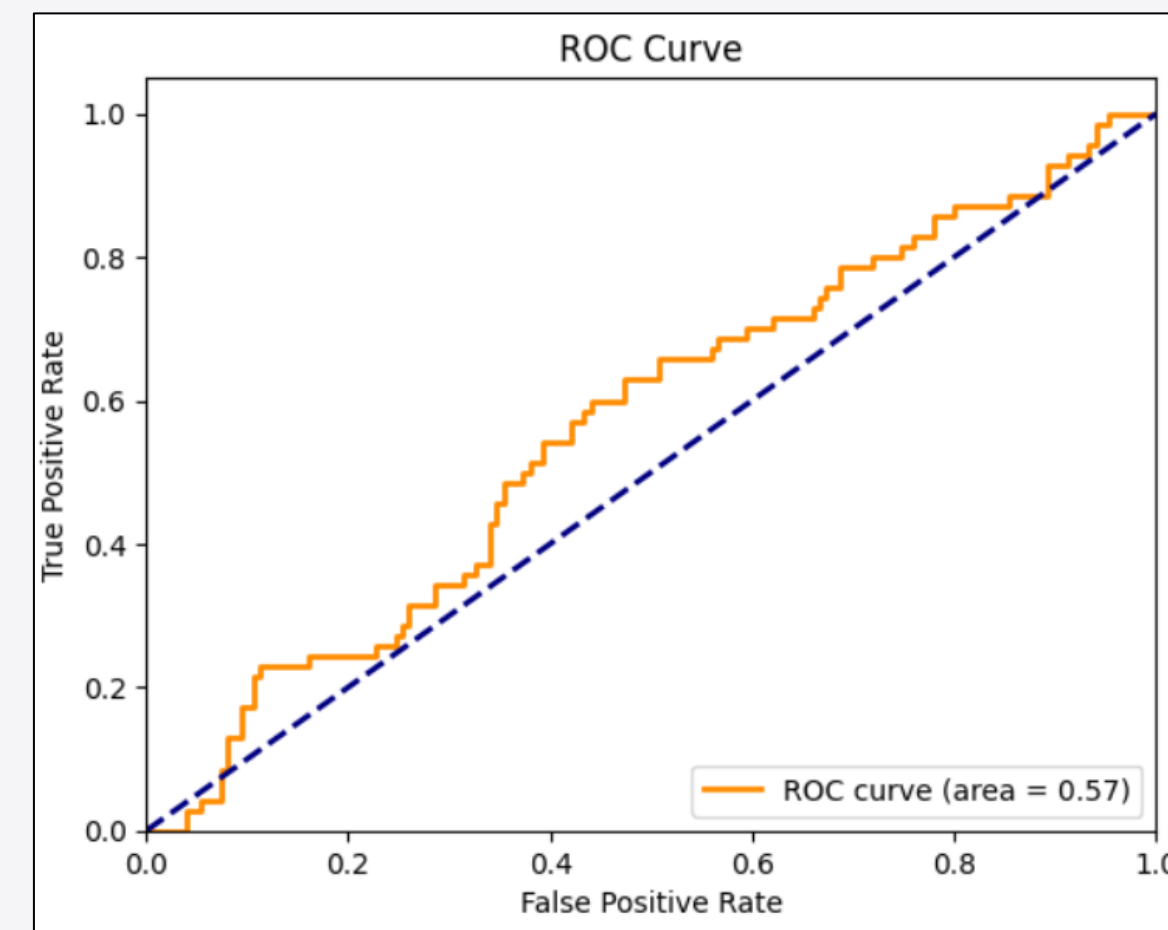
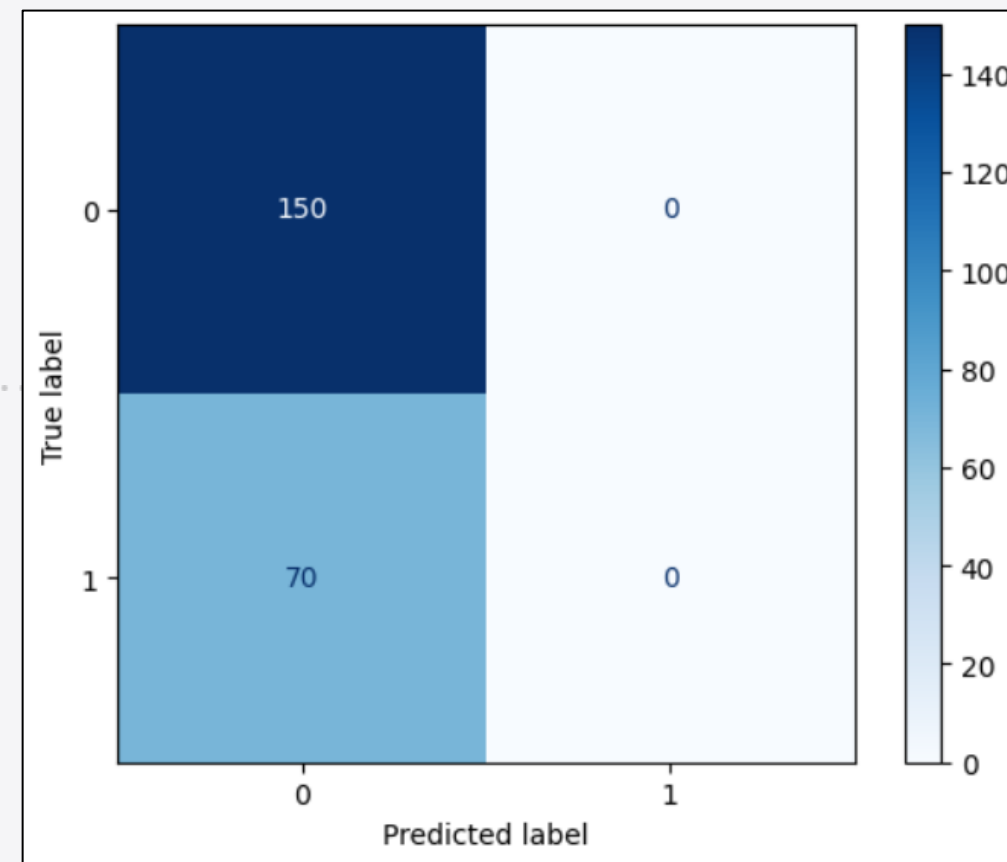


04.

Transfer Learning

1. Logistic Regression

- 문제점: 모든 사진의 결과값을 0으로 예측
 - 사진의 y값이 1일 땐 정확도 0%
- AUC: 약 0.57
 - 찍는 것(AUC=0.5)보다 살짝 나은 수준
- 해결책
 - 7차원 감정분석 벡터값 대신, 4096차원 임베딩 벡터값을 사용하기
 - Logistic Regression 대신 딥러닝 모형 구현하기



04.

Transfer Learning

2. Deep Learning

- 3번의 은닉층을 거침 (활성화함수: ReLU)
 - 4096 → (1) 512 → (2) 256 → (3) 128
- 출력층에선 sigmoid 함수로 확률 계산 후, binary cross-entropy 손실 함수값 계산
- 오버피팅 방지를 위해, 매 은닉층 계산 후 값 중 절반을 dropout
- Adam 옵티마이저 사용 (lr = 0.001)
- 총 20 epochs 훈련

	0	1	2	3	4	5	6	7	8	9	...
0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.000000	...
1	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.000000	...
2	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.000000	...
3	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.000000	...
4	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.000000	...
...
726	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.000000	...
727	0.0	0.0	0.0	0.0	0.0	0.050186	0.0	0.0	0.0	0.003529	...
728	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.000000	...
729	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.000000	...
730	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.000000	...

731 rows × 4096 columns

Model: "sequential_1"

Layer (type)	Output Shape	Param #
dense_4 (Dense)	(None, 512)	2097664
dropout_3 (Dropout)	(None, 512)	0
dense_5 (Dense)	(None, 256)	131328
dropout_4 (Dropout)	(None, 256)	0
dense_6 (Dense)	(None, 128)	32896
dropout_5 (Dropout)	(None, 128)	0
dense_7 (Dense)	(None, 1)	129

=====
Total params: 2262017 (8.63 MB)
Trainable params: 2262017 (8.63 MB)
Non-trainable params: 0 (0.00 Byte)

04.

Transfer Learning

2. Deep Learning

- 더 이상 모든 값을 0으로 예측하지 않음
 - 사진의 y값이 1일 땐 정확도 62.9%
- AUC: 약 0.91
 - 앞선 0.57에 비해 크게 향상
- 한계점
 - y값이 0인 데이터 위주로 구성됨 (497:234)
 - 딥러닝은 black box, 왜 0이고 왜 1인지 판단한 이유를 알 수 없음
 - 컴퓨터 성능 부족으로 은닉층을 더 쌓진 못함 → 전이학습을 사용한 이유

