

서울 아파트, 1년 뒤 올랐을까

당신의 감 vs 당신이 만든 인공지능 두뇌.

국토교통부 실거래가 3,418개 단지로 케라스 두뇌를 만들어, 2025년에 5% 넘게 오른 단지를 맞춥니다.

3,418

서울 단지·면적대
2024년 실제 거래

50%

아무렇게나 찍으면

73%

평당가 한 열만 넣은 두뇌

75%

다 넣어도 여기가 천장

자료: 국토교통부 실거래가 공개시스템 (계약일 기준, 해제 거래 제외) · 과거 데이터로 과거를 맞추는 연습이며 투자 조언이 아닙니다 · Connect AI LAB

왜 이걸 하나

새 모델이 나와도 빠대는 이 여섯 단계다

챗GPT도, 로봇 두뇌도 표를 숫자로 바꾸고 → 두뇌에 넣고 → 가르치고 → 답안지를 쓰는 빠대 위에 있습니다.

오늘 이 여섯 단계를 한 번 익히면, 데이터만 바뀌서 어떤 문제든 같은 손으로 다룰 수 있습니다. 코드를 몰라도 됩니다. 셀마다 무엇을 하는지 적어 두었습니다.

1

명단 내려받기

셀 하나로 자동. 가르칠 명단 3,000개 + 시험 명단 418개

2

글자를 숫자로

구 이름 → 입지 등급, 빈칸 채우기

3

어떤 열을 넣을까

표에서 숫자 덩어리 꺼내기 ← 포인트 ①

4

두뇌 만들기

칸 하나짜리 가장 얇은 두뇌 ← 포인트 ②

5

가르치기

3,000개로 w 와 b 를 정한다 ← 포인트 ③

6

답안지 만들기 · 제출

418개의 답을 파일로 → 순위표에서 채점

데이터

2024년에 거래된 단지가 2025년에 5% 넘게 올랐나

가르칠 명단

3,000개 · 정답 있음

단지·면적대마다 2024년 거래가 가운데 값과 **올랐나(1/0)**. 이걸로 두뇌를 가르칩니다.

시험 명단

418개 · 정답 없음

두뇌가 답안지를 씹니다. 순위표에 올리면 **418개 중 몇 개**를 맞혔는지 나옵니다. 1%는 4개.

왜 5%인가. 2025년 서울은 대체로 올랐기 때문에 「0%보다 올랐나」로 물으면 80%가 정답이라 찍기만 해도 80%입니다.

5% 기준이면 정확히 반반(51%)이라 아무렇게나 찍으면 50%. 두뇌가 배운 만큼만 그 위로 올라갑니다.

2024년 1~12월 거래 → 2025년 1~8월 거래 비교. 단지 · 동 · 구 · 면적대(5m²)로 묶고, 한 해 거래 2건 이상만. 해제된 거래 제외.

명단에 있는 열

두뇌에 넣을 수 있는 숫자들

열	뜻	모양
구·동	어디에 있나	글자
면적	전용면적 m ² . 84면 흔히 말하는 34평	숫자
층·연식	거래들의 가운데 층, 지은 지 몇 년(2024 기준)	숫자
거래수·거래가	2024년에 몇 번, 얼마에(만원, 가운데 값)	숫자
평당가	거래가 ÷ 평수 (만원)	숫자
직전변동	2024년 상반기 → 하반기 몇 % 올랐나	숫자, 573개 빈칸 (반기 거래 없던 단지)
동평당가·구평당가	그 동네 · 그 구의 평당가 가운데 값	숫자
올랐나	2025년 거래가가 5% 넘게 올랐으면 1	가르칠 명단에만

두뇌는 곱하고 더하는 계산만 합니다. 글자와 빈칸이 있으면 계산 자체가 안 되므로 2단계에서 손봅니다.

먼저 사람이 찍는다

당신의 감은 몇 개?

가르칠 명단에서 열 개를 뽑았습니다. 다섯은 올랐고 다섯은 안 올랐습니다. **정답을 가린 채 먼저 찍어 보세요.** 화면에서는 정답 칸을 누르면 보입니다.

#	단지	면적	연식	거래가(만원)	평당가	직전변동	정답
1	금천구 럭키남서울아파트	41m ²	43년	42,000	3,365	-8.7%	안 올랐다
2	용산구 울트라멤버스	132m ²	21년	115,000	2,879	+19.0%	안 올랐다
3	강남구 디에이치아너힐즈	60m ²	5년	222,750	12,312	+11.9%	올랐다 ↑
4	종로구 현대뜨레비앙	28m ²	21년	29,000	3,412	+7.1%	올랐다 ↑
5	노원구 보람아파트	55m ²	36년	47,200	2,858	+4.3%	안 올랐다
6	서초구 방배중앙센트레빌	60m ²	3년	116,500	6,430	+11.1%	올랐다 ↑
7	노원구 신동아(366)	115m ²	31년	119,750	3,432	+5.9%	올랐다 ↑
8	노원구 공릉동한보에센시티아파트	85m ²	24년	64,500	2,511	-4.8%	안 올랐다
9	성북구 동신	60m ²	26년	58,900	3,252	+1.5%	안 올랐다
10	서초구 방배아이파크	126m ²	18년	215,000	5,660		올랐다 ↑

열 개 중 일곱을 넘기면 두뇌와 비슷한 수준입니다. 두뇌는 418개에서 이 비율을 유지합니다. 사람은 보통 못 합니다.

명단 내려받기 — 셀 하나

```
import numpy as np          # 숫자 계산
import pandas as pd         # 표 다루기
import keras                # 두뇌 만들기

!wget -q -O 가르칠명단.csv "https://www.aicitybuilders.com/apt/seoul_apr_train.csv"
!wget -q -O 시험명단.csv "https://www.aicitybuilders.com/apt/seoul_apr_test.csv"

가르칠명단 = pd.read_csv('가르칠명단.csv') # 3,000개, 정답(올랐나) 있음
시험명단   = pd.read_csv('시험명단.csv')   # 418개, 정답 없음
가르칠명단.head()                  # 앞 다섯 줄 보기
```

`pd.read_csv` 가 파일을 읽어 표로 만듭니다. 이제부터 「가르칠명단」은 3,000줄짜리 표입니다. `.head()` 는 잘 읽혔는지 눈으로 보는 용도.

글자를 숫자로 — 뜻이 있는 숫자로

구를 0~24 번호로 바꾸면 「노원구(8)가 강남구(0)보다 크다」는 엉뚱한 뜻이 됩니다.

대신 **강남 3구는 2, 한강 가까운 구는 1, 나머지는 0**. 이걸 「입지」라고 부릅니다. 빈칸은 0으로.

```
입지등급 = {'강남구': 2, '서초구': 2, '송파구': 2,  
            '용산구': 1, '성동구': 1, '마포구': 1, '광진구': 1, '동작구': 1, '강동구': 1, '양천구': 1, '영등포구': 1}
```

```
for 명단 in [가르칠명단, 시험명단]:          # 두 명단에 똑같이  
    명단['입지'] = 명단['구'].map(입지등급).fillna(0)  # 구 → 0 / 1 / 2  
    명단['직전변동'] = 명단['직전변동'].fillna(0)      # 반기 거래 없으면 0
```

가르칠 때 강남구를 2로 넣었으면 시험 볼 때도 2여야 합니다. 그래서 두 명단에 **똑같이** 합니다.

어떤 열을 두뇌에 넣을까

쓸열 = ['평당가', '연식', '면적'] # < 넣어 보세요: '입지', '동평당가', '직전변동', '거래가', '거래수', '층'

입력 = 가르칠명단[쓸열].values.astype('float32') # 표 → 숫자 덩어리 (3000 × 열 개수)

정답 = 가르칠명단['올랐나'].values.astype('float32') # 3000개의 0 / 1

시험입력 = 시험명단[쓸열].values.astype('float32')

평균, 퍼짐 = 입력.mean(axis=0), 입력.std(axis=0) + 1e-6 # 가르칠 명단 기준

입력 = (입력 - 평균) / 퍼짐 # 크기 맞추기 (표준화)

시험입력 = (시험입력 - 평균) / 퍼짐 # 시험 명단에도 같은 값으로

평당가는 수천, 연식은 수십이라 크기가 다릅니다. (값 - 평균) ÷ 퍼짐 으로 비슷한 크기로 맞추칩니다. 쓸열 만 바꾸면 아래는 전부 따라옵니다.

두뇌 만들기 — 칸 하나

```
오름두뇌 = keras.Sequential([
    keras.Input(shape=(len(쓸열),)),          # 쓸 열 개수만큼 들어옵니다
                                # ← 여기에 keras.layers.Dense(16, activation='relu'), 를 넣으면 층이 하나 늘어납니다
    keras.layers.Dense(1, activation='sigmoid'), # 칸 하나. 오를 확률을 냅니다
])
오름두뇌.compile(loss='binary_crossentropy', optimizer=keras.optimizers.Adam(0.003), metrics=['accuracy'])
오름두뇌.summary()                          # Param # = 정해야 할 숫자 개수
```

칸 하나가 하는 일

w 곱하고, b 더하고, 0~1로

열이 3개면 정해야 할 숫자는 **4개**(w 셋 + b 하나). sigmoid 가 그 합을 「오를 확률」로 바꿉니다.

COMPILE

자와 고치는 법

loss 는 틀린 정도를 재는 자, **optimizer** 는 w 와 b 를 고치는 방법, 0.003 은 한 번에 고치는 보폭.

가르치기 — 몇 번 볼까

```
기록 = 오름두뇌.fit(입력, 정답, epochs=100, batch_size=32, validation_split=0.2, verbose=0)
```

```
연습점수 = 기록.history['accuracy'][-1] * 100
```

```
검증점수 = 기록.history['val_accuracy'][-1] * 100
```

```
print(f'연습 점수 {연습점수:.1f}% · 처음 보는 문제 점수 {검증점수:.1f}%')
```

3,000개를 처음부터 끝까지 한 번 보는 게 1 epoch. 100번 되풀이합니다. 20%는 떼어 두고 **처음 보는 문제**로 씁니다.

연습 점수만 오르고 처음 보는 문제 점수가 안 오르면 **외우는 중**입니다. 시험 문제를 미리 본 학생과 같습니다. epochs 를 줄이세요.

노트북 그대로 실행하면 연습 71% · 처음 보는 문제 70% 근처. 둘이 비슷하면 외우지 않은 겁니다.

답안지 만들기 · 제출

```
예측 = (오름두뇌.predict(시험입력, verbose=0).flatten() >= 0.5).astype(int) # 확률 → 0 / 1
```

```
답안지 = pd.DataFrame({'번호': 시험명단['번호'], '이름': 시험명단['이름'], '예측': 예측})
답안지.to_csv('답안지.csv', index=False)
```

```
from google.colab import files
files.download('답안지.csv') # 내 컴퓨터로 내려받습니다
```



제출 전 확인

418개 · 예측이 0과 1뿐 · 번호 3001~3418. 노트북의 확인 셀이 셋 다 ✓ 를 찍어 줍니다.



댓글에 올리기 — 대회 안 해도 됩니다

노트북이 만들어 주는 한 줄을 유튜브 댓글에 그대로. 「처음 보는 문제 70.3% · 쓸열 평당가+연식+면적 · 층 0개 · epochs 100」 남들이 뭘 넣어 몇 %가 나왔는지 서로 보는 게 이 실습의 절반입니다.



순위표에 올리기 — 진짜 점수가 궁금하면

aicitybuilders.com/contest?c=apt 에 로그인하고 답안지.csv 를 올리면 시험 명단 418개 기준으로 바로 채점. 엑셀로 열어 저장하지 말고 그대로.

결과

학습 사다리 — 실제로 재 봤다

두뇌	시험 명단 418개 점수	뜻
아무렇게나 찍기	50%	기준선
평당가 한 열만	73%	비싼 곳이 더 올랐다
로지스틱 3열 (노트북 그대로)	72%	연식·면적은 거의 보태지 못함
로지스틱 9열 전부	74~75%	천장
총 1개(16), 300 epoch	71%	외웠다 — 오히려 떨어짐
총 1~2개 + 조기 종료	73~75%	천장을 못 넘는다

3시드 평균. 같은 코드에서 시작값이 무작위라 실행마다 1~2% 왔다 갔다 합니다. 418개에서 1%는 4개.

이 명단이 가르쳐 주는 것

총을 쌓아도 안 오르는 데이터가 있다

01

비싼 곳이 더 올랐다

평당가 하나로 73%. 나머지 여덟 열을 다 넣어도 2%p. 2025년 서울은 **한 줄**로 대부분 설명됩니다.

02

두뇌가 나쁜 게 아니다

천장 75%는 두뇌의 한계가 아니라 **데이터에 규칙이 그만큼만** 있어서입니다. 나머지는 잡음.

03

외움의 신호

연습 점수는 오르는데 처음 보는 문제 점수가 안 오르면 잡음까지 외우는 중. **총을 줄이거나 epochs 를 줄인다.**

04

다음 회차 — 주식

같은 코드로 「내일 오르나」를 시키면 **52%** 동전과 같습니다. 배울 규칙이 시장에 없기 때문입니다. 그걸 직접 확인합니다.

세 군데를 바꾸고 3 → 4 → 5 → 6 을 다시 실행

1

열 넣기

`쓰열` 에 '동평당가', '입지', '직전변동' 을 넣어 보고, 반대로 '평당가' 하나만 남겨 보세요.

2

층 쌓기

4단계 화살표 자리에 `keras.layers.Dense(16, activation='relu')`, . 올랐나요, 떨어졌나요? 떨어졌다면 연습 점수와 처음 보는 문제 점수의 차이를 보세요.

3

덜 외우기

epochs 를 30으로, 또는 `callbacks=[keras.callbacks.EarlyStopping(patience=10, restore_best_weights=True)]` . 처음 보는 문제 점수가 나빠지기 시작하면 멈춥니다.

!

막히면

NameError 위 셀 건너뛰, 1번부터 · **KeyError: '입지'** 2단계 안 함 · **ValueError shape** 3단계 바꾼 뒤 4단계 다시 · 점수 50% 근처면 아무것도 못 배운 것, 2·3단계 다시

정리

같은 코드, 다른 데이터, 다른 천장

서울 아파트 75% · 주식 52% · 규칙이 많은 데이터는 90% 넘게. 두뇌는 같고 천장이 다릅니다. 천장을 정하는 건 데이터 속 규칙의 양입니다.

이걸 몸으로 아는 사람은 새 모델이 나와도 불안하지 않습니다.

코랩 노트북 내려받기

순위표 · 진짜 점수

이 두뇌에 지식을 붙이는 과정

노트북은 코랩에서 「파일 → 노트 업로드」로 엽니다 · 몇 %가 나왔는지, 뭘 넣었는지 댓글로 올려 주세요 · Connect AI LAB · AI CITY BUILDERS · 2026-09-08 · 국토교통부 실거래가 공개시스템 자료, 투자 조언 아님