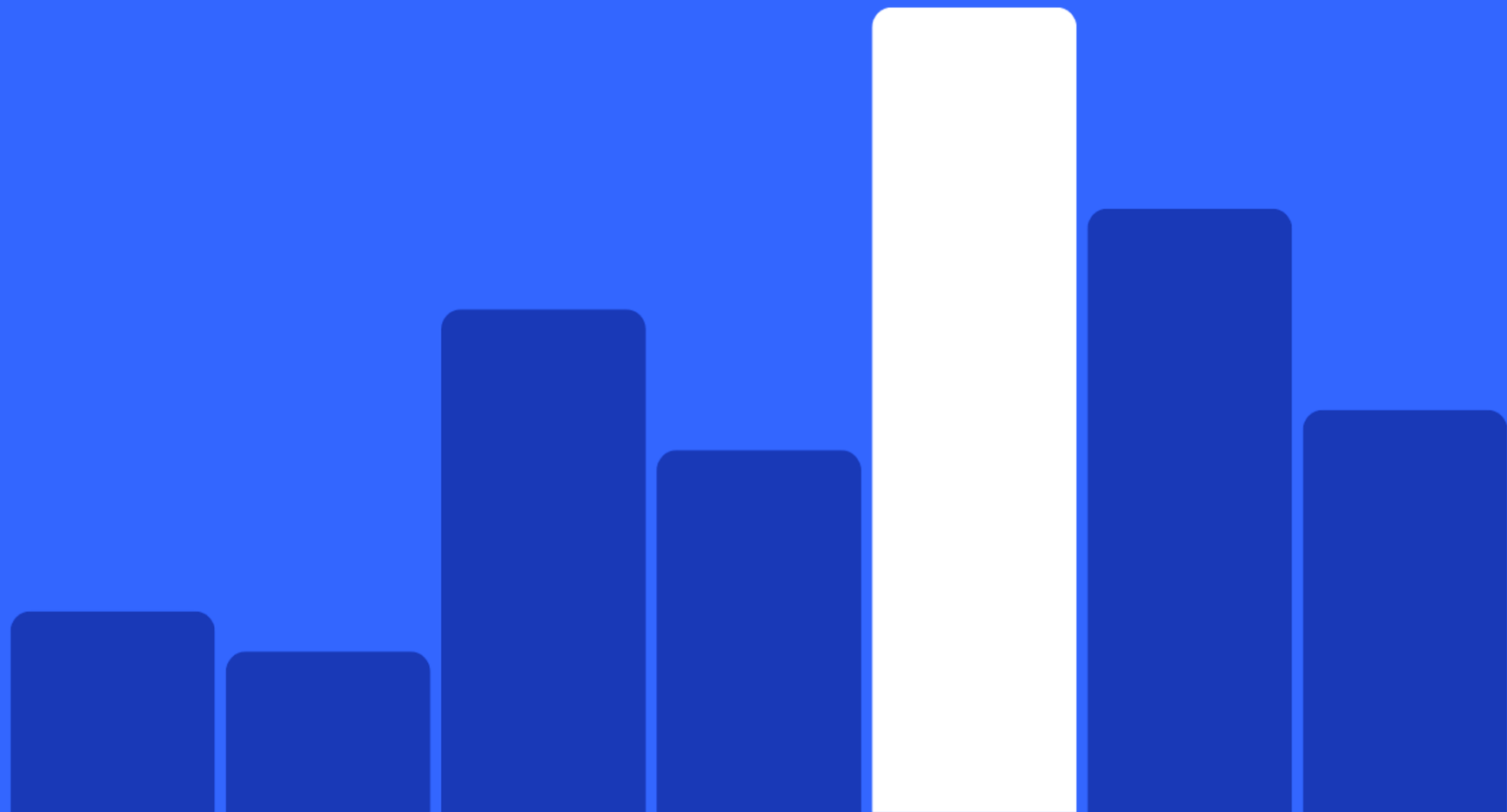
머신 러닝으로 만드는 연봉예측 계산기

인공지능 개인별 과제

2021108244 고다영



목차

개요

연봉이라는 요소는 그저 개인의 수익이라고 느낄 수도 있지만, 다르게 보자면 스스로가 기업체에 하는 기여를 평가받은 것이라고도 볼 수 있습니다. 그리고 대다수의 경우, 근속 기간이 길어질 수록 우리는 업무에 보다 능숙해지고, 보다 많은 일을 해내며 그 기여를 키워가곤 합니다. 그러한 경향을 통해 현대 사회에서는 승진에 따라 연봉을 지급하는 '직무급'외에도 근로자들의 근속 연수를 기반으로 연봉을 지급하는 '연공급'적인 연봉 체계가 상당히 오랜 시간 동안 큰 비중으로 자리해 왔습니다. 본 프로젝트에서는 이러한 경향을 기반하여 근속 연수를 통한 연봉 예측의 정확도를 높이는 것을 목표로 한다.

이미지 출처 :https://www.news700.kr/1546

개요

당연하게도, 금전은 인간의 생활상과 큰 관련이 있고 그만큼 많은 관심을 받았다. 임금 상승, 연차에 따른 연봉 증가에 대한 다양한 연구와 도구가 있었다.
이에 대하여 조사하여 확인한 어느 논문에서는, 임금의 인상에 대한 계산식을 작성한 바도 있었다.
하지만 이 경우 요구 데이터가 너무 다종다양해 명백히 연구용에 해당하여, 대중이 일반적인 연봉의 상승폭을 알아보기에는 번거로워 부적합한 면이 있다. 하여 나는 보다 간결하고 적은 데이터를 기반으로 대략적인 연봉을 예측하는 프로그램을 작성하고자 하였다.

$$\ln W_{ijt} = \beta 근속연수_{ijt} + \gamma 승진_{ijt} + AX_{ijt} + BZ_{jt} + CYD_t + \delta_j + u_{ijt}$$

lnW_{ijt} = j 기업 근로자 i 의 t 년도 연간 총 근로소득의 로그값 X 100,
$근속연수_{ijt}$ = j 기업 근로자 i 의 t 년도 근속연수
β = 근속연수의 회귀계수
$승진_{ijt}$ = j 기업 근로자 i 의 t 년도의 승진 횟수
γ = 승진의 회귀계수
X_{ijt} = j 기업 근로자 i 의 t 년도 근로자 특성 벡터
A = 근로자 특성의 회귀계수 벡터
Z_{jt} = j 기업의 t 년도 기업 특성 벡터
B = 기업 특성의 회귀계수 벡터
YD_t = t 년도 가변수 벡터
C = 년도 가변수의 회귀계수 벡터,
δ_j = j 기업의 고정효과
u_{ijt} = 잔차

자료 1 : 임금의 연공성은 왜 생기는가?: 사무직의 근속연수-임금 관계에서 연공에 의한 승진의 매개효과를 중심으로(박희준)

데이터 살펴보기

Experience Salary Dataset

이 데이터 세트에는 다양한 산업 분야의 직원들의 업무 경험(개월)과 해당 월급(천 달러) 간의 관계에 대한 정보가 포함되어 있습니다. 그것은 데이터 애호가와 야심찬 데이터 과학자들이 경험을 바탕으로 급여 예측을 분석하고 모델링함으로써 선형 회귀 기술을 연습할 수 있도록 설계되었습니다.

# exp(in months)	# salary(in thousan...
Experience of employee in month	Salary of employee from different industries in thousand dollars
 0.7 49.5	 -1.28 55.3
18.29029331840369	16.52182490535894
17.023406691171306	11.666233723936466
26.343613477194726	23.167254546085978
19.105833871730955	20.877145423479412
27.742516466299612	23.166235691979292

데이터 살펴보기

Experience Salary Dataset

이 데이터 세트에는 다양한 산업 분야의 직원들의 업무 경험(개월)과 해당 월급(천 달러) 간의 관계에 대한 정보가 포함되어 있습니다. 그것은 데이터 애호가와 야심찬 데이터 과학자들이 경험을 바탕으로 급여 예측을 분석하고 모델링함으로써 선형 회귀 기술을 연습할 수 있도록 설계되었습니다.

```
print(data.head())
```

```
   exp(in months)  salary(in thousands)
0       18.290293              16.521825
1       17.023407              11.666234
2       26.343613              23.167255
3       19.105834              20.877145
4       27.742516              23.166236
```

```
print(data.columns)
```

```
Index(['exp(in months)', 'salary(in thousands)'], dtype='object')
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 2 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   exp(in months)        1000 non-null   float64
 1   salary(in thousands)  1000 non-null   float64
dtypes: float64(2)
memory usage: 15.8 KB
```

데이터 살펴보기

데이터 전처리

exp(in years) : 근속 기간을 연 단위 기준으로 저장. float형태

exp를 만들어 근속 기간을 반년 단위로 반올림하여 저장. float형태이며 0.5의 배수이다.

이를 상위 10번째 까지 출력하였다.

```
data['exp(in years)'] = data['exp(in months)'] / 12
# exp(in years)를 0.5 단위로 저장
data['exp'] = (data['exp(in years)']*2).round()*0.5
print(data.head(10))
```

```
   exp(in months)  salary(in thousands)  exp(in years)  exp
0       18.290293             16.521825       1.524191  1.5
1       17.023407             11.666234       1.418617  1.5
2       26.343613             23.167255       2.195301  2.0
3       19.105834             20.877145       1.592153  1.5
4       27.742516             23.166236       2.311876  2.5
5       31.671171             32.966251       2.639264  2.5
6       14.186399             15.294170       1.182200  1.0
7       29.932845             33.159461       2.494404  2.5
8       32.841327             32.032653       2.736777  2.5
9       26.873869             32.347784       2.239489  2.0
```

데이터 시각화

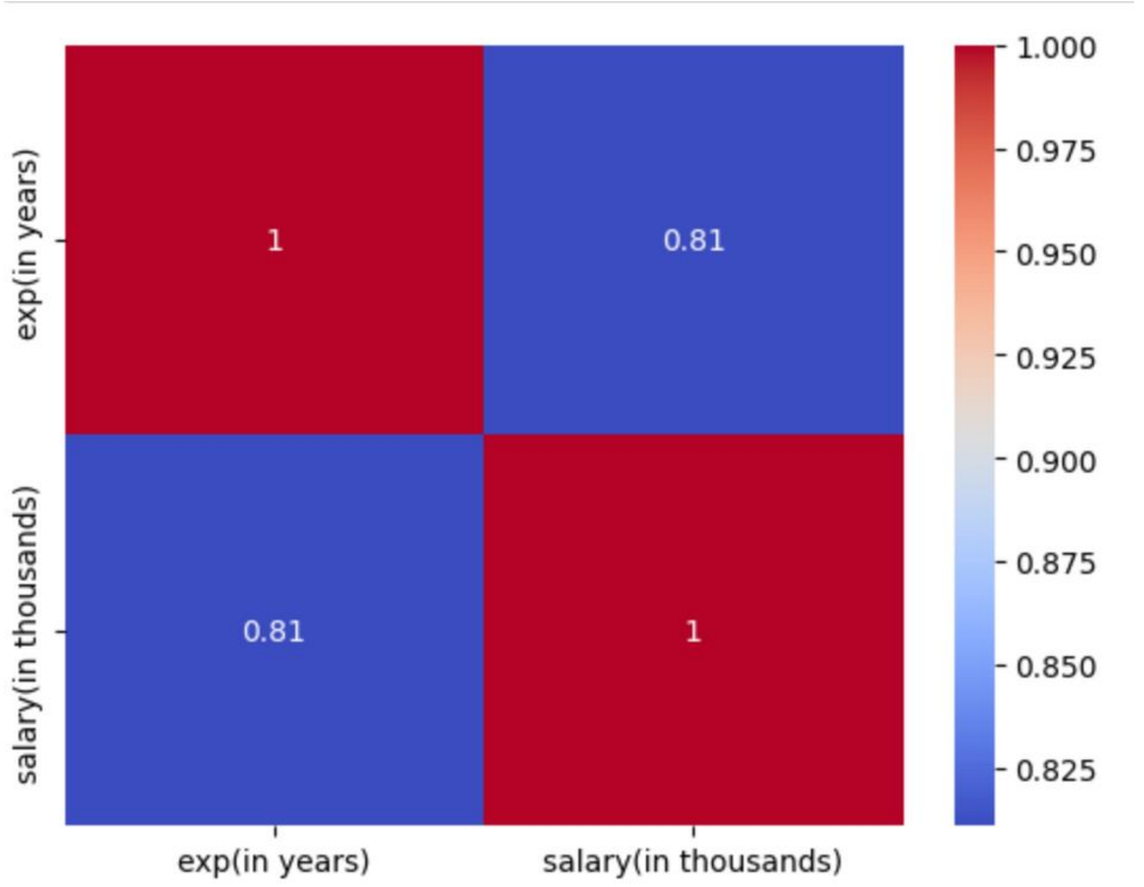
scattered형태의 plot을 이용하여 출력.

X축 : 근속일(연간)

Y축 : 연봉(천 달러)

Scattered Chart형태

데이터 시각화



히트맵

히트맵은 여러 요소 간에 연관성을 파악하는 데 도움이 된다. 다만, 이 경우는 단 2개의 열을 사용하기에 큰 영향을 주지 않는다.

exp(in years)와 salary(in thousands)간의 연관은 0.81이다.

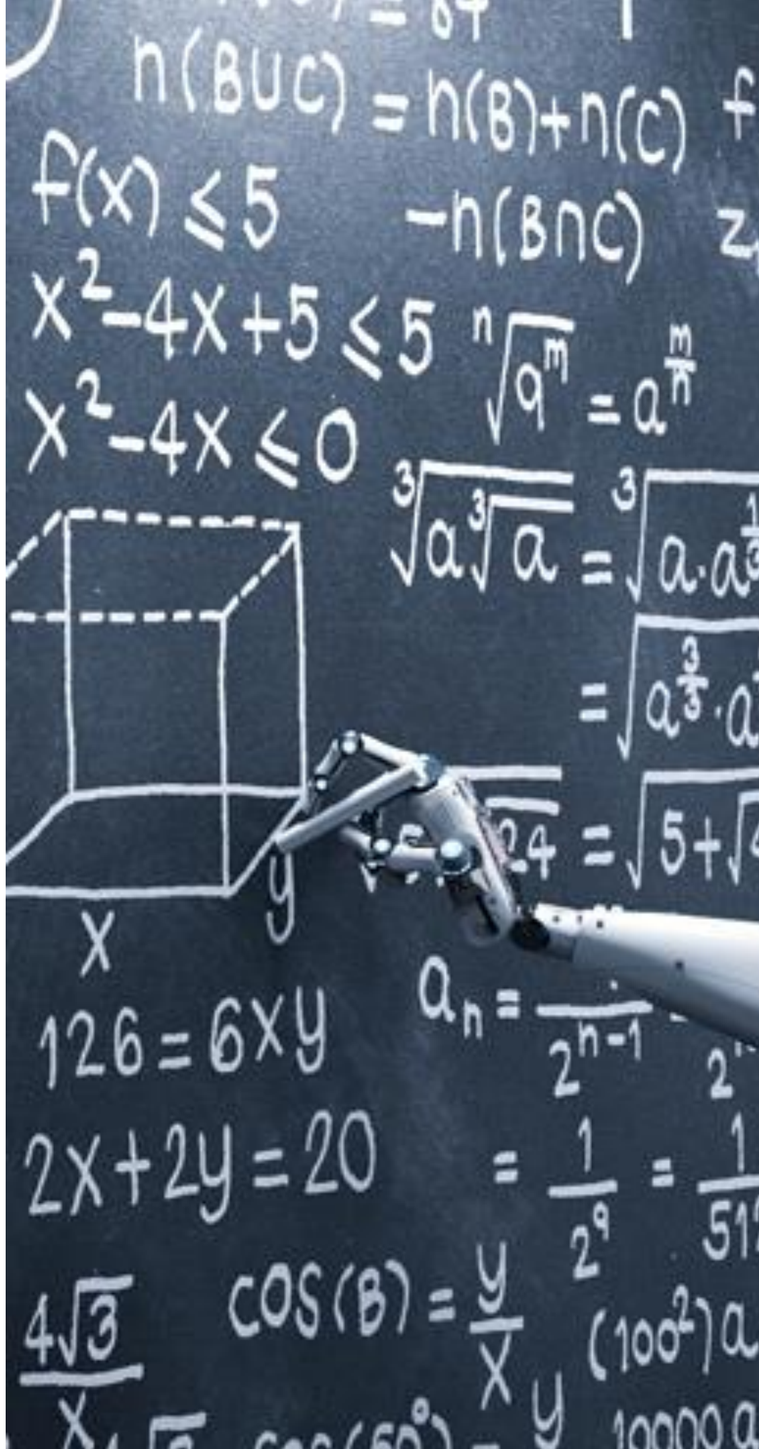
딥러닝 모델 비교

데이터 학습 모듈

퍼센티지 상 정확도로는 Linear Regression Model Score가 최고
Decision Tree Model의 경우 0.33점대로 정확도 매우 낮음
하지만 기본적으로 정확도가 떨어진다. 임금에 영향력을 갖는 다른 요소를 추가하면 좋을 듯 하다.

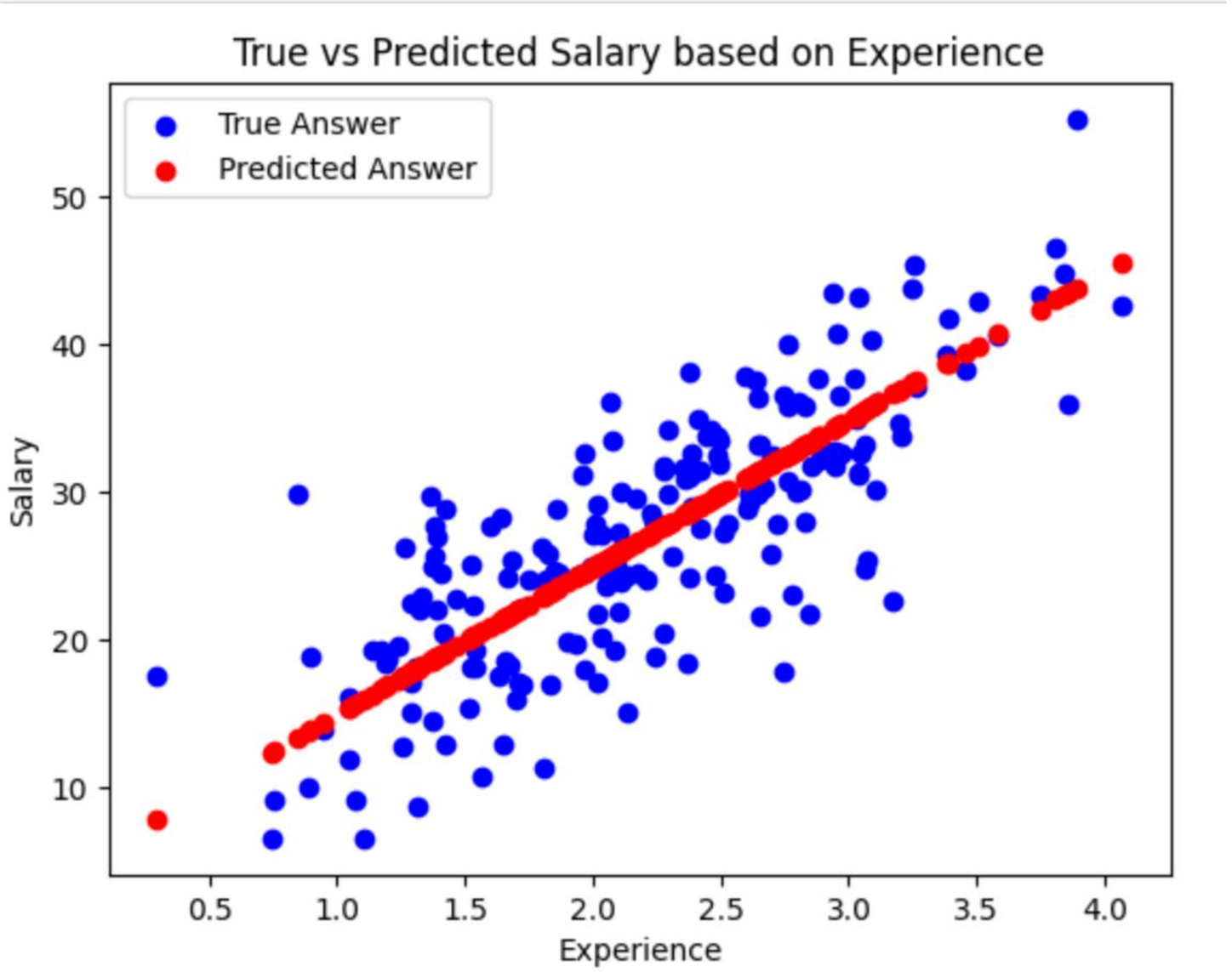
```
for i, model in enumerate(models):
    model.fit(X_train, y_train)
    print(f'{model_names[i]} Model Score: {model.score(X_test, y_test)}')
```

```
Decision Tree Model Score: 0.33877922692465334
K-Nearest Neighbors Model Score: 0.6117379841114977
Linear Regression Model Score: 0.6208258933084713
Random Forest Model Score: 0.5226367658993465
```



Linear-Regression-Model

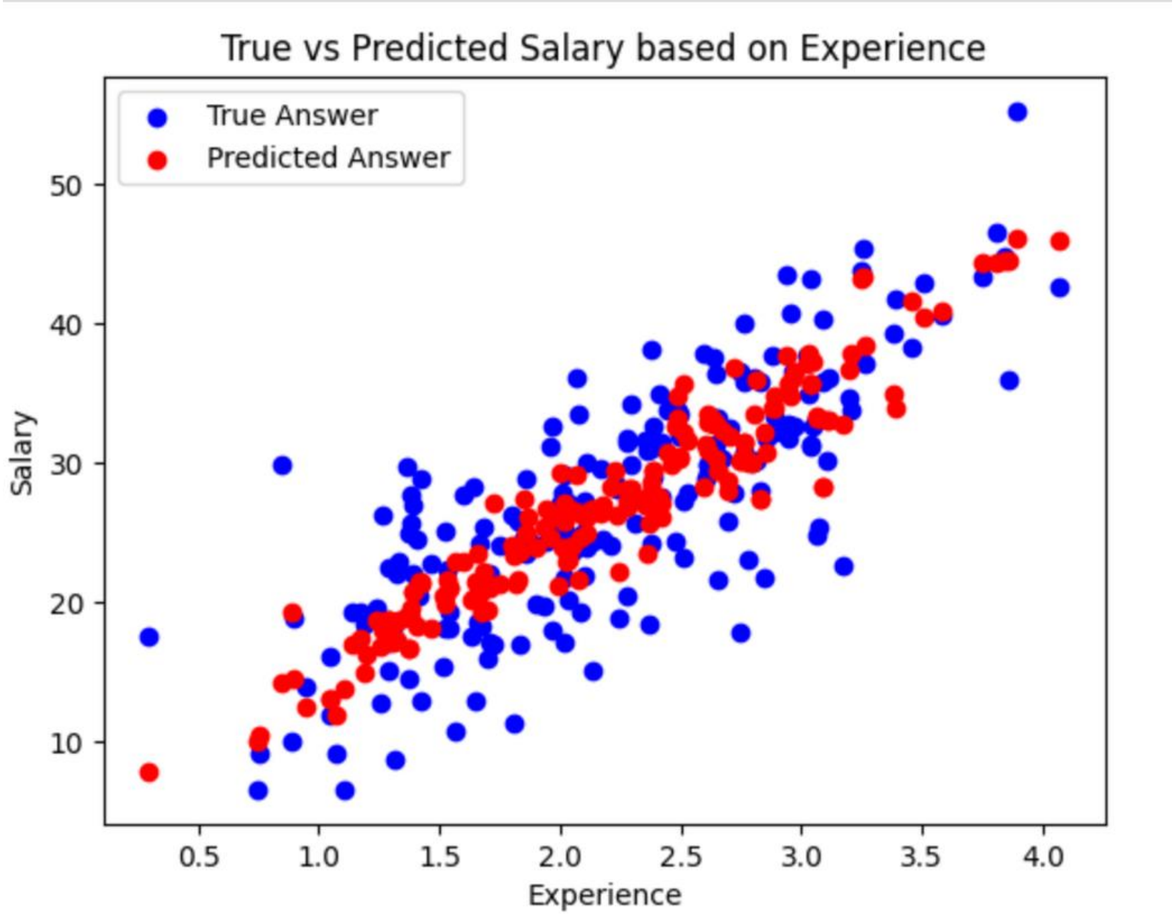
Score: 0.6208258933084713



선형 회귀 방식의 인공지능 데이터학습은 중도의 값을 갖는 그 특성상, 확실히 안정도가 높았다. 다만 지나치게 일률적인 면이 있어 어느정도 가깝게 맞추는 능력과 별개로 정확히 맞는 값을 내는 것은 어려워보인다. 보다 다양한 인자로 정확한 예측을 낸다면 다른 결과를 얻을 지도 모르지만 현재로서는 알 수 없다.

K-Nearest-Neighbors-Model

Score: 0.6117379841114977



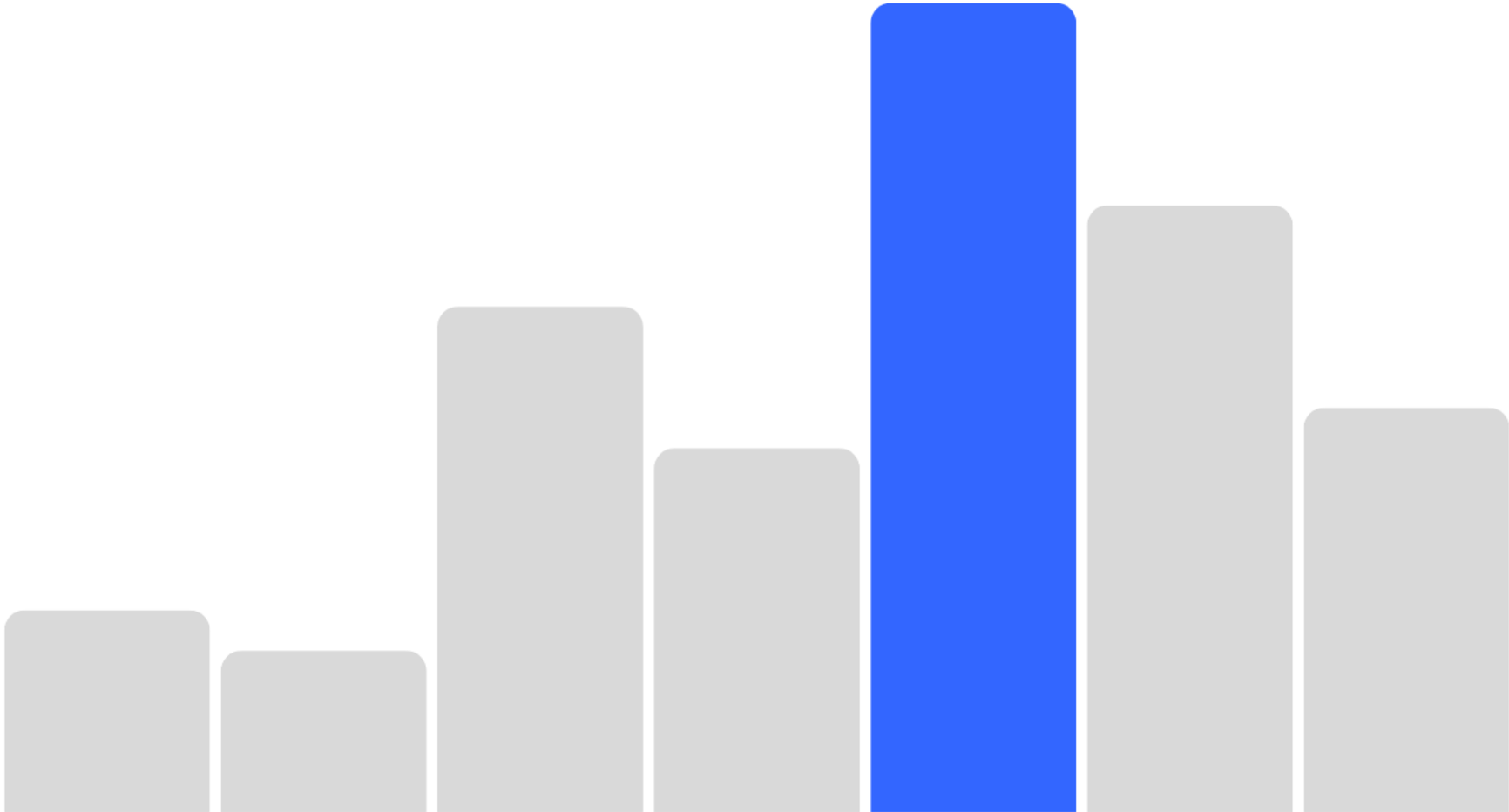
K 최근접 이웃 모듈은 예측 값이 정답 값과 비슷한 형태로 어느정도 분산되어 있다는 점이 선형 회귀와는 다른 점이다. 하지만 그 스코어가 선형 회귀보다 미미하게 떨어지고, 또한 값이 크게 튈 수 있다는 불안요소를 안고 있다.

결론

보다 적절한 자료 사용의 필요가 여실히 드러났다. 일단 해외의 자료라는 점에서 국내의 실정과는 차이가 필연적으로 생기고, 자료에 대략적인 직종도 표기되지 않아 직종 간 연봉의 편차가 클 수 있다는 점이 정확도를 크게 낮추었다. 또한 단순히 개개의 연봉 뿐 아니라 인물이 구분되어 인상치를 측정할 수 있다면 더 좋았을 것이라 생각한다. 데이터를 다루는 데 있어 용도에 적합한 구성의 데이터셋을 선정하는 능력이 필요하다고 느꼈다.
이후 추가적으로 더 적합한 데이터를 찾아 프로젝트를 재조정해볼 여지가 있다.

감사합니다!

AI 개인별 리포트



2021108244 고다영