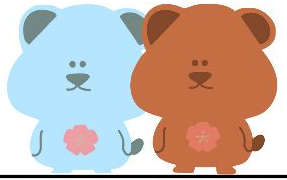
데이터 사이언스 직군의 연봉 결정 요인

팀 paylytics: 오지후, 이덕주, 이재린, 함예린

EWHA EURON

목차

- 서론
- 본론
  - Week.2
  - Week.3
  - Week.4
  - Week.5
- 결론



EWHA EURON

서론

#서론

프로젝트 목표: 데이터 사이언스 직군 연봉 예측에 최적화된 회귀 모델 찾기

선정한 데이터: Jobs and Salaries in Data Science
(https://www.kaggle.com/datasets/hummaamqaasim/jobs-in-data/data)

- 변수 총 12개 (work_year, job_title, job_category, salary_currency, salary, salary_in_usd, employee_residence, experience_level, employment_type, work_setting, company_location, company_size)

종류	변수명
숫자형 (int)	work_year, salary, salary_in_usd
문자형 (object)	job_title, job_category, salary_currency, employee_residence, experience_level, employment_type, work_setting, company_location, company_size

Week.2

데이터 분석 방향 확정 및 가설 설정

EWHA EURON

#01 데이터 선정

1. 초기 주제: **공공데이터 기반 맞춤형 자격증 추천 모델 선정**

2. 공공 데이터 활용하기 어려움

   - api 키 발급, 데이터 전처리

3. 데이터 및 주제 변경

   - 주제: **데이터 사이언스 직군 연봉의 결정요인 분석**

   - 데이터: Jobs and Salaries in Data Science

   (https://www.kaggle.com/datasets/hummaamqaasim/jobs-in-data/data)

#02 데이터 1차 전처리

1. 데이터 작성 연도 (work_year)

- 2023년에 집중된 데이터
+ 동일한 작성 연도 데이터를 비교하는 것이 정확할 것으로 판단

➔ **work_year가 2023인 데이터만 추출**

2. 직업 분류 (job_category)

- 속성이 비슷한 job_title & job_category

- job_title은 세분화된 경향이 있음

➔ job_title 드랍, **job_category**를 분석에 활용

EWHA EURON

#03 가설 설정

1. 연봉 결정과 상관성이 높은 요인을 찾기 위함
2. 연봉과 각 변수의 관계 분석

분석 요인 (변수)	가설 설정
직업 분류 (job_category)	연구직, 관리직 및 경영, 기술 및 설계직으로 분류한 뒤 연봉과의 상관관계 파악
경력 (experience_level)	경력과 급여의 상관관계 파악 - 경력이 많을수록 급여가 높아지는 경향이 있다
기업 크기 (company_size)	기업 크기와 급여의 상관관계 파악 - 급여: 중견기업 > 대기업 > 중소기업

3. [피드백] 급여 예측 정확도 개선

    - 하나의 변수 활용 < 여러 변수 활용

➔ **방향성 수정!**

Week.3

데이터 전처리 확정 및 가설 검증

EWHA EURON

#데이터 2차 전처리

1. 추가적인 칼럼 생성

job_group - job_category를 분류
- 분석: Data Analysis, BI and Visualization
- 관리 및 운영: Leadership and Management, Data Management and Strategy, Data Quality and Operations, Cloud and Database
- 데이터 처리 및 모델 개발: Data Science and Research, Data Engineering, Machine Learning and AI, Data Architecture and Modeling

분석: 0, 관리 및 운영:1, 데이터 처리 및 모델 개발:2

#데이터 2차 전처리

2. 이상치 작업

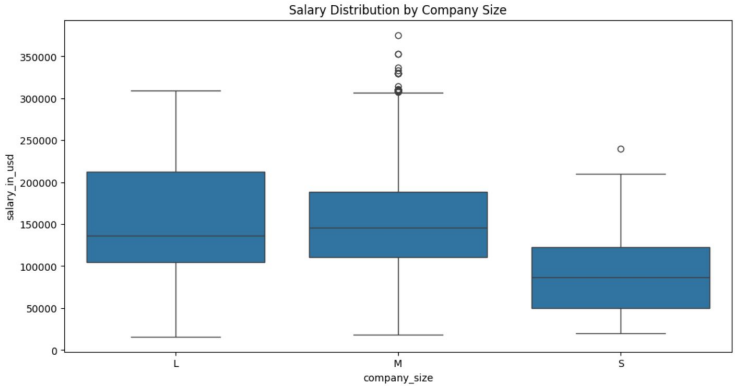
- 직급(experience_level)별로 이상치 추정해서 작업 진행

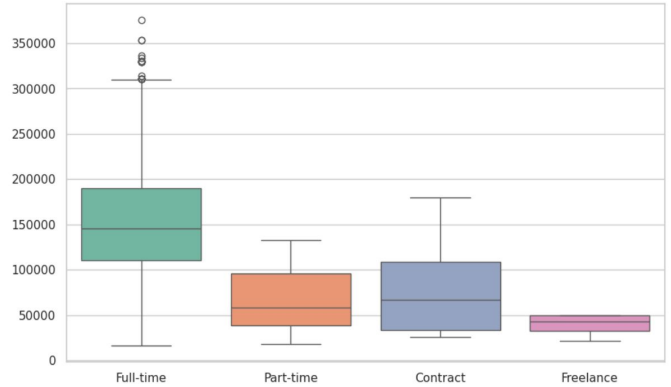
3. 일부 칼럼 제거

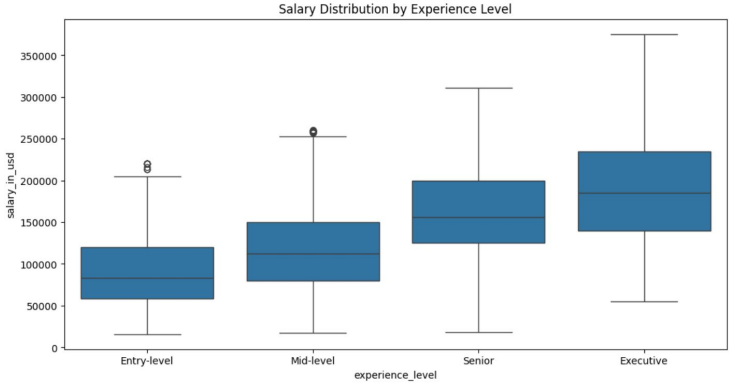
- 연봉: salary, salary_currency 칼럼 제거
- 위치: employee_residence 칼럼 제거
  - company_location 칼럼과 너무 유사하기 때문!!

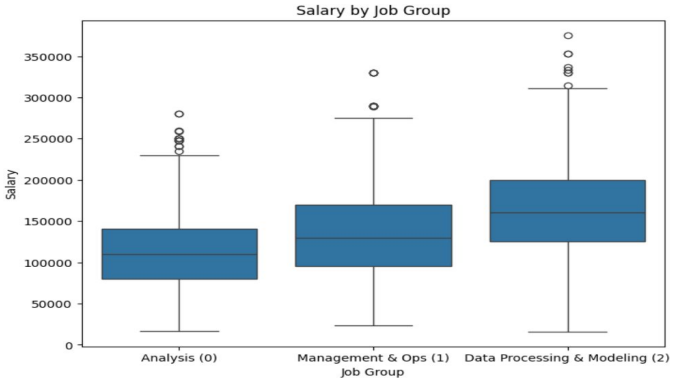
#가설 검증

- 기업 크기, 경력, 고용형태, 직무









Week.4

피드백 반영

EWHA EURON

#방향성 설정 관련

문제	피드백	해결
프로젝트 자체가 일반 분석으로 치우치는 경향 => 세션에서 배운 모델을 적용하기가 어려움	목표를 세워라 문제를 다시 정의 => 모델링을 진행한다면 해당 모델링을 통해 어떤 것을 증명하고 싶은가?	연봉 예측을 목표로 모델링 진행 => 회귀 모델을 활용하기로 결정

!!연봉 예측에 최적화된 회귀 모델을 찾자!!

#데이터 전처리 관련

문제	피드백	해결
범주형 데이터가 많음 => Label Encoding과 One-Hot Encoding을 각각 언제 적용해야 하는가?	Label Encoding은 크기 정보가 잘못된 의미로 학습될 수 있음 One-Hot Encoding은 범주의 수가 많을 경우 피처가 폭발적으로 늘어나는 문제 발생	선형 모델 : One-Hot Encoding 적용 트리 기반 모델 : One-Hot Encoding 및 Label Encoding 적용

Week.5

모델링

EWHA EURON

#모델링

1. 데이터 전처리

데이터 전처리

삭제
: salary
: salary_currency
: employee_residence
: work_year
(2023 데이터 필터링 후 drop)

유지
: job_group

데이터 스케일링

타겟변수
: 로그변환 적용 유무 비교

나머지 피처
: 정규화 하지 않음

Encoding

선형모델
: one-hot encoding

트리기반모델
: one-hot encoding, label encoding

#모델링

2. 모델링 역할 분담

데이터 전처리

- 오지후

: 전처리 후 파일 배부

선형모델

- 함예린

: linear, ridge, lasso, elasticnet

트리기반모델

- 이재린, 이덕주, 오지후

(재린)
: randomforest,gradient boosting

(덕주)
: decisiontree,baggingregressor

(지후)
: catboost,xgboost,lightgbm

❯ **성능평가지표: RMSE, R^2, MAE, MSE**

EWHA EURON

#모델링

3. 모델링 결과_ 선형모델

★ 데이터 전처리: one-hot encoding

```
print('get_dummies() 수행 전 데이터 Shape:', df.shape)
cols=["job_title", "job_category", "employment_type", "work_setting", "company_location","experience_level", "company_size","job_group","companyLoc_group"]
df_ohe=pd.get_dummies(df, columns=cols, drop_first=True)
print('get_dummies() 수행 후 데이터 Shape:', df_ohe.shape)
```

#모델링

3. 모델링 결과_ 선형모델

★ 모델 학습 예측 평가

	Linear regression	Ridge	Lasso	ElasticNet
최초값	조정할 하이퍼 파라미터가 없음	MSE: 2542396163.567 RMSE: 50422.179 MAE: 39337.397 Variance Score: 0.345	MSE: 2530413525.855 RMSE: 50303.216 MAE: 39191.110 Variance Score: 0.348	MSE: 2583009725.302 RMSE: 50823.319 MAE: 39687.590 Variance Score: 0.334
GridSearch CV		최적의 알파 값: 10.0000 최적의 R2: 0.3289		Best Params: {'alpha': 0.0021209508879201904, 'l1_ratio': 0.7000000000000001, 'max_iter': 5000} 최적의 R2: 0.331

EWHA EURON

#모델링

3. 모델링 결과_ 선형모델

★ 모델 학습 예측 평가

	Linear regression	Ridge	Lasso	ElasticNet
Ridge/ Lasso/ Randomized SearchCV		최적의 알파 값: 3.727593720314938 최적의 R2: 0.349	최적의 알파 값: 10.0 최적의 R2: 0.349	Best Params: {'alpha': 0.20594494295802446, 'l1_ratio': 0.9729188669457949} 최적의 R2: 0.320 👎
최종값	**MSE: 2540296788.552** **RMSE: 50401.357** **MAE: 39295.708** **R2: 0.345**	**MSE: 2527669166.916** **RMSE: 50275.930** **MAE: 39196.164** **R2: 0.349**	**MSE: 2525762437.357** **RMSE: 50256.964** **MAE: 39131.135** **R2: 0.349**	**MSE: 2583009725.302** **RMSE: 50823.319** **MAE: 39687.590** **R2: 0.334**

#모델링

3. 모델링 결과_ 트리모델(randomforest,gradientboosting)

★ 데이터 전처리

- ➔ LabelEncoding: "experience_level", "company_size” 칼럼
- ➔ One-Hot Encoding: 나머지 피처
- ➔ 정규화 X

#모델링

3. 모델링 결과_ 트리모델(randomforest,gradientboosting)

★ 모델 학습 예측 평가

	RandomForestRegressor	GradientBoostingRegressor
최초값	MAE: 38392.007308671644, MSE: 2348973276.2795506, RMSE: 48466.20757063163, R2 Score: 0.35205973843300054	MAE: 38933.35706049112, MSE: 2367798652.898577, RMSE: 48660.03136968345, R2 Score: 0.34686695076963936
	- 이상치 제거, - Feature Selection (feature_importances≤0인거 제거)	- 이상치 제거, - Feature Selection (feature_importances≤0인거 제거)

EWHA EURON

#모델링

3. 모델링 결과_ 트리모델(randomforest,gradientboosting)

★ 모델 학습 예측 평가

	RandomForestRegressor	GradientBoostingRegressor
하이퍼 파라미터 튜닝	'max_depth': 12 'min_samples_leaf': 2 'min_samples_split': 5 'n_estimators': 200	'learning_rate': 0.05 'max_depth': 3 'n_estimators': 300 'subsample': 0.8
최종값	MAE: 38198.48401744415 MSE: 2310440687.798335 RMSE: 48067.04367649768 R2 Score: 0.3626885589953682	MAE: 38431.11709846258 MSE: 2310182312.5925508 RMSE: 48064.3559469234 R2 Score: 0.3627598291541686

#모델링

3. 모델링 결과_ 트리모델(decisiontree,baggingregressor)

★ 데이터 전처리

➔ One-Hot Encoding: decisiontree

➔ LabelEncoding: baggingregressor(문자형-> 숫자형 컬럼 변환)

#모델링

3. 모델링 결과_ 트리모델(decisiontree,baggingregressor)

★ 모델 학습 예측 평가

	decisiontree	baggingregressor
최초값	MAE: 39279.3207 MSE: 2562181319.6944 RMSE: 50617.9940 R^2: 0.3296	MAE: 39237.8540 MSE: 2543884491.9359 RMSE: 50436.9358 R^2: 0.3208
GridSearch CV	Fitting 5 folds for each of 7 candidates, totalling 35 fits GridSearchCV 최고 평균 MSE: 2769189179.4152 최적 하이퍼파라미터: {'max_depth': 12}	최적 하이퍼 파라미터: {'max_depth': 8, 'min_samples_leaf': 8, 'min_samples_split': 8, 'n_estimators': 100} 최고 예측 MSE: 2742335746.4604

#모델링

3. 모델링 결과_ 트리모델(decisiontree,baggingregressor)

★ 모델 학습 예측 평가

	decisiontree	baggingregressor
하이퍼파라미터 튜닝 강화	개선된 모델 성능: MAE: 40381.4448 MSE: 2686968108.3219 RMSE: 51835.9731 R^2: 0.2969 👎	튜닝된 랜덤 포레스트 성능 평가: MAE: 39332.6898 MSE: 2507583261.7884 RMSE: 50075.7752 R^2: 0.3305
특성 중요도 확인 및 사용	불필요한 특성 제거 후 성능: MAE: 40943.7751 MSE: 2761730239.5279 RMSE: 52552.1668	중요 특성 기반 성능 평가: MAE: 39450.8194 MSE: 2569240337.9218 RMSE: 50687.6744 R^2: 0.3141

#모델링

3. 모델링 결과_ 트리모델(catboost,xgboost,lightgbm)

★ 데이터 전처리: One-hot encoding

★ 모델 학습 예측 평가

	xgboost	lightgbm	catboost
최초값	R2: 0.33 mse: 2508767232.00 rmse: 50087.60 mae: 38857.29	R2: 0.34 mse: 2473541345.09 rmse: 49734.71 mae: 38687.01	R2: 0.33 mse: 2496543645.51 rmse: 49965.42 mae: 39205.95
GridSearch CV		GridSearchCV 최적 파라미터: {'colsample_bytree': 0.5, 'max_depth': 5, 'min_child_weight': 1	
최종값		R2: 0.34 mse: 2478726266.97 rmse: 49786.81 mae: 38803.10	

EWHA EURON

결론

EWHA EURON

#결론

RandomForestRegressor의 성능이 가장 좋다!

➔ 연봉을 예측하기에 가장 적합

THANK YOU

EWHA EURON