



Mission 15: 학습 자동화 및 예측 분석 보고서



Docker Hub URL

<https://hub.docker.com/r/yena1/mission15-train>



데이터 개요 및 EDA 요약

본 프로젝트에서는 학생의 학습 관련 데이터를 기반으로 **성취도(Performance Index)**를 예측하는 모델을 구축하였다.

7,000개의 데이터와 6개 열로 구성되어 있음, 주요 피쳐는 다음과 같다.

- 파일: `researcher1/eda_modeling.ipynb`
- 타겟: `Performance Index` (성취도 점수)
- 주요 인사이트:
 - `Previous Scores` 성취도와 매우 높은 상관관계 ($r=0.91$)
 - `Hours Studied` 중간 정도 영향 ($r=0.37$)
 - 나머지 변수들은 영향 미미
- 가설:
 - 이전 점수가 높을수록 성취도 상승
 - 공부 시간이 많을수록 성취도 상승
 - 수면시간/과외활동은 영향 미미



모델링 결과 요약



모델 구성 및 학습 개요

- 전처리 구성:

- 수치형 피쳐(`Hours Studied` , `Previous Scores` , `Sleep Hours` , `Sample Question Papers Practiced`)
- 범주형 피쳐(`Extracurricular Activities`) → `OneHotEncoder`
- `ColumnTransformer` 로 통합 후 파이프라인 구성
- 데이터 분할: 학습 80%, 검증 20%
- 모델 후보: `Ridge Regression` , `Random Forest Regressor`

하이퍼파라미터 튜닝

모델	주요 탐색 파라미터	최적 조합	RMSE(CV)
Ridge	$\alpha=[0.01\sim100]$, solver=['auto','saga']	$\alpha=1.0$, solver='saga'	2.0510
Random Forest	n_estimators=[100~300], max_depth=[5~20], min_samples_split=[2,5,10], max_features=['sqrt','log2',None]	n_estimators=300, max_depth=10, min_samples_split=10	2.2815

최종 모델 선택

학습 데이터 기준 RMSE 비교 결과, **Ridge Regression(RMSE:2.0510)**이 **Random Forest(RMSE: 2.2815)** 보다 낮은 오차를 보여 최종 모델로 채택하였다.

검증 데이터 기준 **Ridge Regression**의 RMSE는 **2.0106** 값이 나왔다.

- 핵심 변수 영향:
 - `Previous Scores` (+17.54)
 - `Hours Studied` (+7.41)
 - `Sleep Hours` , `Activities` → 영향 미미

코드 아키텍처 도식 및 설명

전체 구조 개요

```
researcher1/ → 학습 자동화
├── data/                # 학습용 데이터
│   ├── mission15_train.csv
│   └── mission15_test.csv
├── shared/              # 학습 결과 저장 폴더
│   └── [timestamp]/model.pkl
├── train.py              # 데이터 전처리 + 하이퍼파라미터 튜닝 + 모델 학습 스크립트
├── requirements.txt      # 모델 학습에 필요한 라이브러리 목록
├── Dockerfile            # 학습 환경 정의 파일
└── researcher2/ → 추론 및 분석
    ├── docker-compose.yml # 연구자 1 이미지와 Jupyter 환경을 동시에 실행
    ├── inference.ipynb    # model.pkl 불러와 mission15_test.csv 추론 및 분석
    ├── shared/            # 학습 결과(model.pkl) 자동 공유 폴더
    │   └── [timestamp]/model.pkl
    └── data/              # Docker Hub 이미지에서 전달받은 데이터
```

동작 흐름 요약


연구자1: 학습 자동화 Docker 이미지 생성

1 환경 및 목적

- 목적

학습 과정을 완전히 자동화하여, 컨테이너 실행 시 데이터 로드 → 모델 학습 → 결과 저장까지 일괄 수행

- 환경

 기반 경량 이미지로 구성

- 자동 실행 구조

컨테이너 실행시 `ENTRYPOINT ["python", "train.py"]` 로 학습 자동 수행

2 주요 동작 흐름

단계	내용	설명
① 데이터 로드	<code>mission15_train.csv</code>	<code>/app/data</code> 경로에서 CSV 로드 후 DataFrame 생성
② 전처리 파이프라인 구성	<code>ColumnTransformer</code>	- 수치형(<code>StandardScaler</code>) + 범주형(<code>OneHotEncoder</code>) 통합 전처리
③ 데이터 분할	<code>train_test_split</code>	학습:검증 = 8:2 비율로 분리
④ 모델 정의	Ridge / RandomForest	두 가지 모델을 <code>Pipeline</code> 으로 구성
⑤ 하이퍼파라미터 탐색	<code>GridSearchCV</code>	- Ridge : α , solver 튜닝- RandomForest : 트리 수, 깊이, 분할 조건 등 탐색
⑥ 성능 비교 및 최적 모델 선택	RMSE 기준	교차검증 결과가 가장 낮은 모델을 <code>best_model</code> 로 선정
⑦ Feature Importance 시각화	Ridge: <code>coef_</code>	상위 영향 변수 도출 및 bar plot 시각화
⑧ 모델 저장	<code>joblib.dump()</code>	<code>/app/shared/[timestamp]/model.pkl</code> 로 저장 (KST 기준 폴더명)

연구자2: 추론 및 분석 환경

1 환경 및 목적

- 목적

연구자1이 Docker Hub에 업로드한 학습 이미지를 활용해, `model.pkl` 을 불러와 예측 (`inference`)과 결과 분석을 수행한다.

- 환경

`jupyter/base-notebook` 기반 컨테이너 환경에서 Jupyter Lab을 실행한다.

- 자동 실행 구조

- `docker-compose.yml` 을 통해 두 컨테이너를 연동한다.
 - `data-container` : 연구자1 이미지(`yena1/mission15-train:latest`)
 - `jupyter` : 추론용 환경으로 실행

2 주요 동작 흐름

단계	내용	설명
① 모델 불러오기	<code>model = load(model_path)</code>	<code>/app/shared</code> 에서 학습된 <code>model.pkl</code> 로드
② 테스트 데이터 로드	<code>mission15_test.csv</code>	<code>/app/data</code> 경로에서 CSV 파일 로드
③ 예측 수행	<code>preds = model.predict(X_test)</code>	학습된 Ridge 모델을 활용한 성취도 예측
④ 결과 저장	<code>pd.DataFrame(...).to_csv(result_path)</code>	<code>/app/result/result_[timestamp].csv</code> 저장
⑤ 통계 분석	<code>describe()</code> , <code>mean()</code> , <code>std()</code> 등	예측값의 분포 확인 (평균·표준편차·최솟값·최댓값)
⑥ 피쳐 영향 분석	<code>coef_</code> 기반	Ridge 모델의 가중치(Feature Weight) 시각화
⑦ 시각화 및 해석	<code>matplotlib</code> , <code>seaborn</code>	예측 결과 히스토그램, 상관관계 히트맵 등 출력

✓ 결론

- 최적 모델: Ridge Regression (Validation RMSE = 2.0106)
- 핵심 영향 변수: `Previous Scores` , `Hours Studied`
- 의의:
 - 학습-추론 파이프라인이 완전히 자동화된 Docker 환경에서 재현 가능
 - 연구자 간 모델 공유 및 결과 분석이 용이한 협업 구조 구현
- 한계 및 개선 방향:
 - 입력 피쳐가 단순하여 모델의 예측 다양성은 제한적
 - 향후 학습 태도, 시간대, 과목별 점수 등 **비선형 특성 확장** 시 Random Forest나 XGBoost 계열 모델로 성능 향상 기대 가능