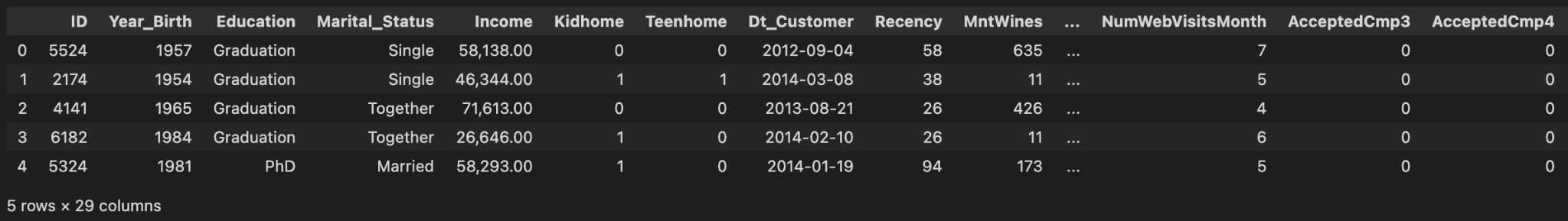
**고객 정보에 따른 구매 여부 예측**

컴퓨터공학과 18101246 이사빈

**1. 분석할 데이터에 대해 설명하고, 해당 데이터 분석의 목적을 서술하시오.**

**Information:**

* Range Index: 2240 entries
* Data columns: total 29 columns (ID, Education, Income, … , Response)
* Data Types: float64(1), int64(25), Object(3)



**해당 데이터 선택 이유**

* 적당한 Size (Underfitting은 피할 수 있을 정도이지만, 학습이 너무 오래걸리지 않아 여러가지 예제에 적용함으로써 다양한 실험을 해볼 수 있음)
* Regression, Classification 모두 적용해볼 수 있는 Target이 존재
* Preprocessing을 통해 수업시간에 배운 내용을 적용해볼 여지가 있음
* 다양한 Feature가 존재하기 때문에, 해당 데이터의 분석을 통해 해석할 요소가 많음

**분석을 통해 얻는 의미**

* 수업시간에 배운 내용을 적용해봄으로써 보다 능동적인 이해
* + 고객의 특성에 따른 마케팅 전략 수립
* + 마케팅 전략에 따른 고객의 반응 예측

**과제의 목표**

* 최대한 수업시간에 배운 내용 및 실습한 library code를 나만의 방식으로 적용해보는 것
* 기본적인 모델(Linear Regression, Logistic Regression 등)을 통해 고객의 반응 예측
* 각 Feature들이 결과 예측에 어떤 영향을 미치는지 분석
* Preprocessing, Regularization 등을 통한 Vanilla(Naive) 모델의 성능 향상
* 데이터 분석 흐름을 이해하기 위한 insight 도출

**Source:**

* Kaggle
* URL : <https://www.kaggle.com/datasets/rodsaldanha/arketing-campaign?datasetId=178248&sortBy=voteCount>

**2. 데이터를 분석하기 위한 파이썬 코드를 작성하고 해당 코드 전체 동작에 대해 간략히 설명하시오.**

**소스코드 구조**

18101246\_term\_project

├── marketing\_raw.csv

├── marketing\_preprocessed.csv

├── preprocessing.ipynb

├── Classification.ipynb

├── Regression.ipynb

**preprocessing.ipynb**

**outline**

* Data Parsing
* Data Information
* Create Several Variables
  + category로 dtype 변경
  + Age, Spending, Children 생성
* Plot some Variables
* Outliers
* High Leverage
* Missing Value
* Correlation
* Export preprocessed dataset

**Classification.ipynb**

**outline**

* Data Fetching and Cleaning
* Classification
  + Logistic regression with lbfgs
* Score
* Best Subset Selection
* Regularization
  + Ridge normalization
* Boosting
* PCA

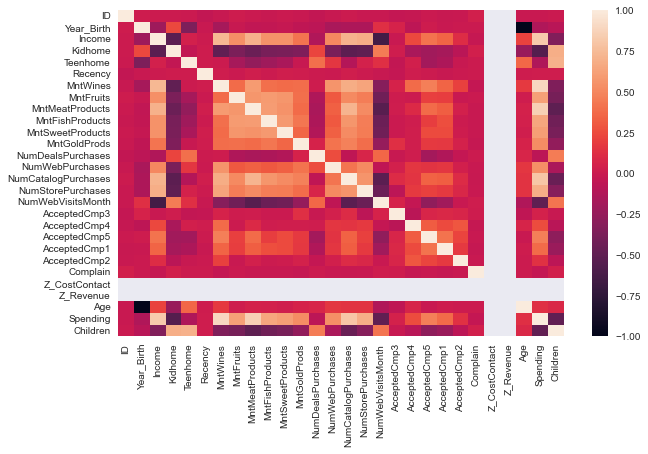
**Regression.ipynb**

**outline**

* Data Fetching and Cleaning
* Regression
  + Ordinary least squares regression
  + ```Income ~ Education + Spending + Children```
* Score
* Second order polynomial
  + Ordinary least squares regression
  + ```Income ~ Education + Spending + Children + Income:Education + Income:Spending + Income:Children```

**3. 파이썬 코드를 통해 실행한 결과들을 바탕으로 데이터 분석 결과를 서술하시오.**

**변수간의 correlation**

****

**Highest 3 by Income**: Spending(0.82), MntWines(0.73), NumCatalogPurchases(0.71)

**Lowest 3 by Income**: NumWebVisitsMonth(-0.65), Kidhome(-0.53), Children(-0.35)

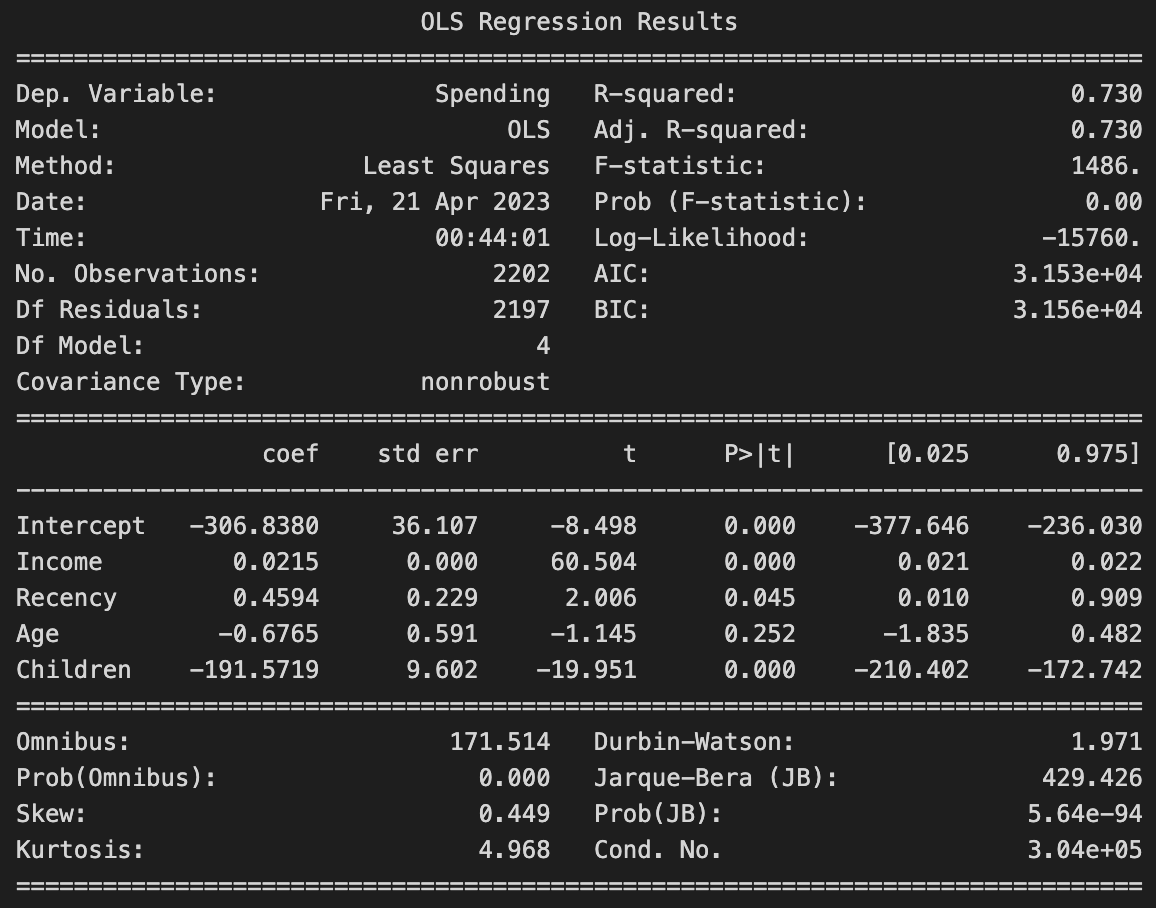
Income이 높을 경우 고가치 상품 소비가 증가하며, 총 소비량이 증가하는 모습을 보인다.

반면, market에 방문하는 횟수가 줄어들며, 자녀가 있을 경우에 음의 상관관계를 따른다.

**OLS Regression Results**

**R formula:** Spending ~ Income + Recency + Age + Children

**결과:**

****

**R-Squared**: 0~1의 값을 가지며, 1에 가까울 수록 데이터를 잘 설명하는 것을 의미한다. 해당 Regression의 경우 0.730의 값을 얻었다. 즉, Dependent variables의 Variance 중 73%가 설명되며, 나머지 27%는 model에 의해 설명되지 않음을 의미한다.

**Coef**: 각 Coefficient는 해당 variable의 영향력을 나타낸다. Income, Recency는 Spending에 양의 관계를, Intercept, Age, Children는 음의 관계를 가짐을 파악할 수 있다.

**T-statistic**: coefficient가 0일 때의 standard error를 계산하여 해당 변수의 유의성을 파악하는 검정에 사용된다.

**P-value**: t-statistic과 직접적인 관계를 가진다, 예를 들어 alpha(significant level)가 0.05일 때, Age의 경우 0.252의 p-value를 가지므로 통계적으로 significant하지 않다는 것을 나타낸다.

**Skewness**: distribution이 좌우 대칭일 경우 0을 갖는데, 해당 model의 경우 0.449를 갖는 것을 보아 오른쪽으로 distribution이 치우쳐져있는 것을 알 수 있다.

**Kurtosis**: distribution이 중앙에 집중되어있는 정도와 꼬리 부분의 두꺼운 정도를 나타낸다. outlier를 사전에 제거했음에도 4.968의 값을 갖는 것을 보아 꼬리가 두꺼운 분포임을 알 수 있다.

**4. 위 파이썬 코드에서, 수업시간에 실습한 library code 및 내용들이 어떻게 활용되고 반영되었는지 설명하시오.**

**sklearn.model\_selection.train\_test\_split()**: 주어진 데이터셋을 8(train):2(test)의 비율로 데이터셋을 분리하기 위해 해당 메소드를 사용하였으며, 매 실험마다 동일한 결과를 얻기 위해 random\_state parameter를 사용한다.

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

**sklearn.linear\_model.LogisticRegression()**: Classification을 위해 간단한 형태의 model을 사용한다. 이후에 Regularization 기법을 적용했을 때 혹은 Cross-Validation을 적용했을 때와 비교하기 위해 추가적인 parameter는 설정하지 않았으며 학습 결과를 시각화했다.

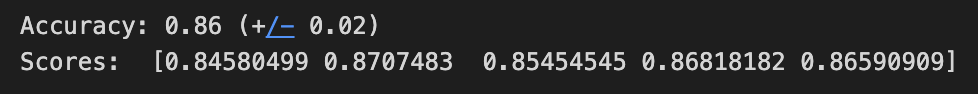
model = LogisticRegression(max\_iter=10000, solver='lbfgs', random\_state=42)

**sklearn.model\_selection.cross\_val\_score():** 모델의 generalization 능력을 향상시키기 위해 사용한다. K=5로 설정한 채로 Score를 계산하여 mean과 std를 확인했다

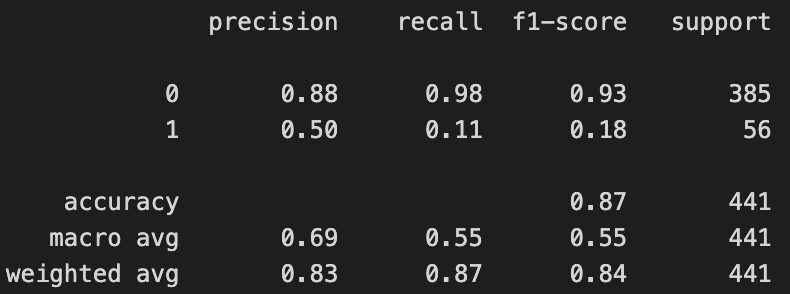
scores = cross\_val\_score(model, X, y, cv=5)

print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), scores.std() \* 2))

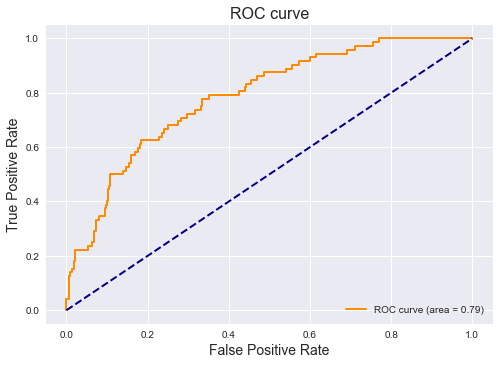
print("Scores: ", scores)



**sklearn.metrics.classification\_report()**: classification score를 여러 metrics으로 시각화하는데 사용한다. f1-score, label의 구성비율 등을 한번에 파악하기 좋다



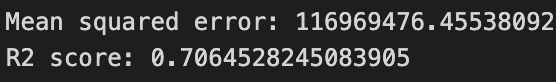
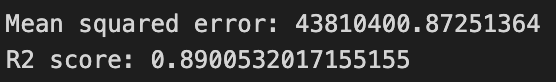
**sklearn.metrics.roc\_curve(), auc()**: binary classification model의 성능을 시각화하는데 사용한다. 좌측 상단으로 그래프가 가까울 수록 좋으며 다음과 같은 결과를 얻었다. / Area = 0.79

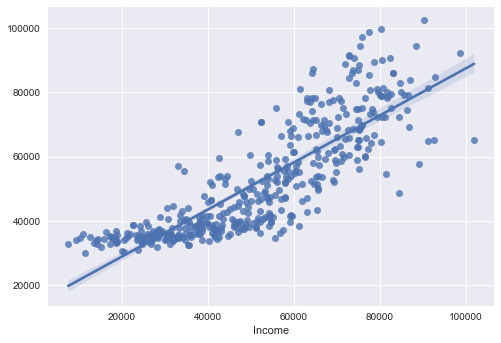
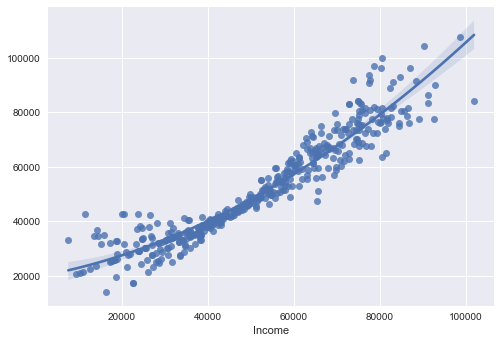


**statsmodels.formula.api.ols()**: Ordinary Least Squares regression을 위한 함수, 3번 지문에서 적용예시를 작성하였다.

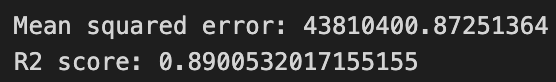
**polynomial features**: 같은 데이터셋에 대해 1차식과 2차식의 결과에 대해 비교하였다.

1차식: 2차식:

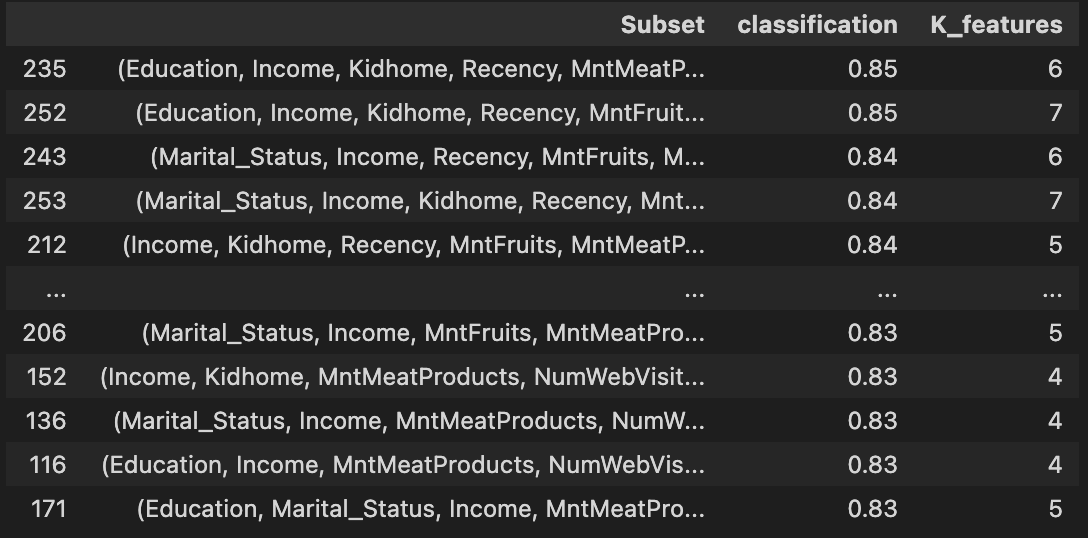
 

**mean\_squared\_error, R2\_score, Accuracy** : Regression의 예측 성능을 평가하는 metric.



**Best Subset Selection**: Feature의 조합에 따라 성능이 달라질 수 있는 점을 고려해 일부 feature들을 대상으로 실제 점수가 바뀌는지 실험하였다.



**Regularization**: Logistic Regression 수행 시 inverse of regularization strength를 나타내는 Parameter Cs를 10으로, penalty는 ridge normalization을 의미하는 l2를 설정하여 model을 학습하였다.

model = LogisticRegressionCV(Cs=10, cv=5, penalty='l2', max\_iter=10000, solver='lbfgs', random\_state=42)

**sklearn.ensemble.BaggingClassifier()**: n\_estimatiors(number of base in estimators in the ensemble)를 100으로 설정하여 학습을 진행하였다.

model = BaggingClassifier(estimator=LogisticRegressionCV(Cs=10, cv=5, penalty='l2', max\_iter=10000, solver='lbfgs', random\_state=42), n\_estimators=100, random\_state=42)

**PCA**: Dimensional Reduction의 개념으로 이해하고 적용해 보았다. overfitting을 방지하기에는 유리하지만, underfitting될 여지가 있으며 같은 조건에서 실행한 BaggingClassifier에 비해 classification 성능 자체는 저조한 모습을 보였다.

pca = PCA(n\_components=2)

X\_train\_pca = pca.fit\_transform(X\_train)

X\_test\_pca = pca.transform(X\_test)