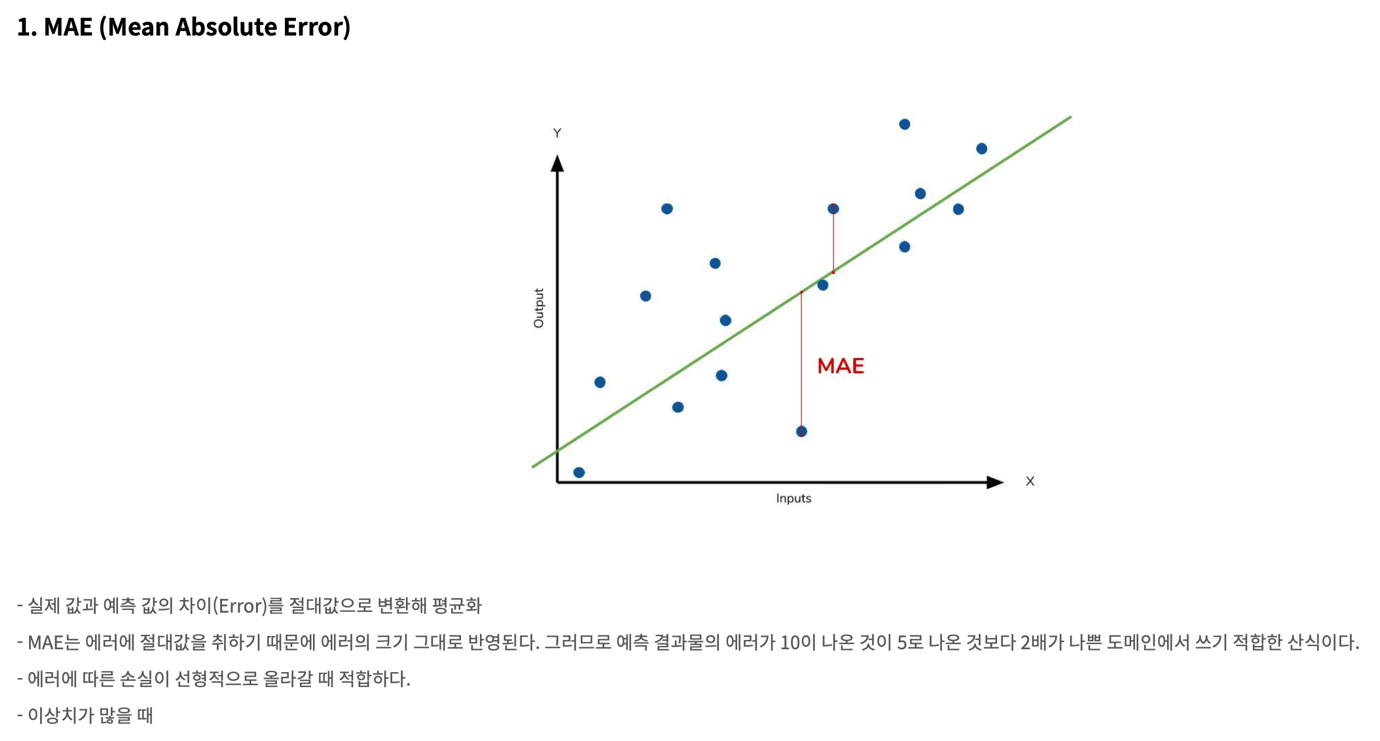
1. **MAE**



#### RMSE (Root Mean Squared Error)

- MSE 값은 오류의 제곱을 구하므로 실제 오류 평균보다 더 커지는 특성이 있어 MSE에 루트를 씌운 RMSE 값을 쓴다.

1. **Df.corr**

테이블이(가) 표시된 사진

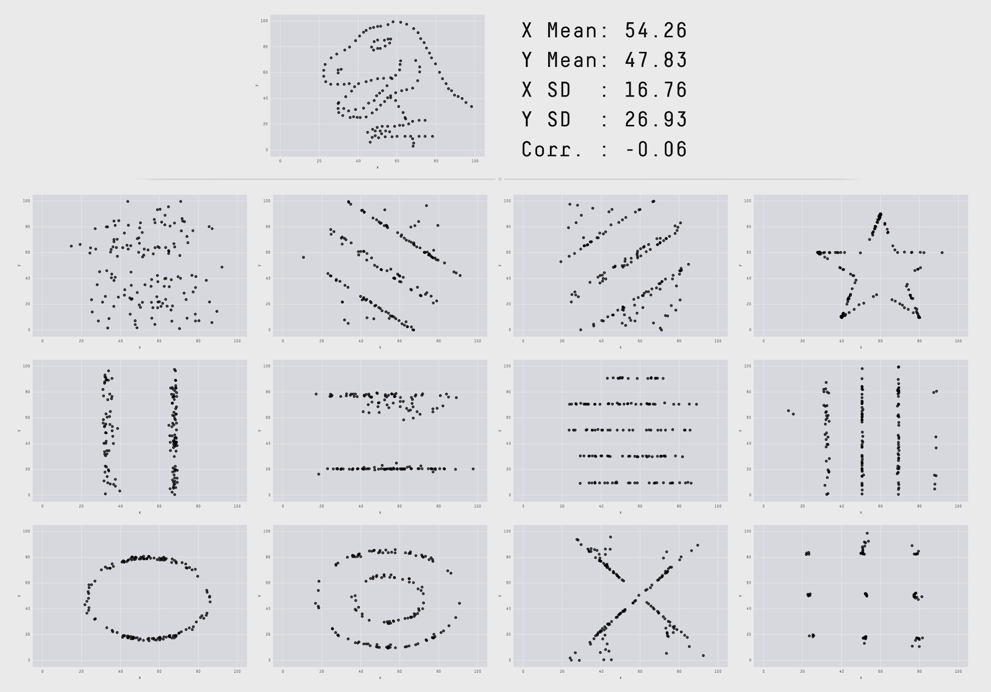
자동 생성된 설명

1. **왜 시각화를 하지**

**1. 많은 양의 데이터를 한눈에 볼 수 있다**

### 2. 데이터 분석에 대한 전문 지식이 없어도, 누구나 쉽게 데이터 인사이트를 찾을 수 있다

### 3. 요약 통계보다 정확한 데이터 분석 결과를 도출할 수 있다



 소수점 두 자릿수 기준으로 동일한 요약 통계를 갖는 12개의 데이터 셋을 시각화했을 때 시각적 패턴이 모두 다르다는 것을 알 수 있습니다.

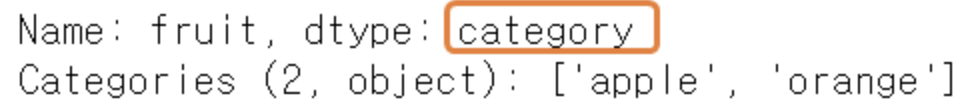
### 5. 데이터 시각화를 활용할 수 있는 분야와 방법이 무궁무진하다

1. 텍스트이(가) 표시된 사진

   자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

1. 텍스트이(가) 표시된 사진

   자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

train["AgeCategoryCode2"] = train.AgeCategory.map({"Young":0, "Middle":1, "Old":2})

1. 텍스트이(가) 표시된 사진

   자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

n\_jobs=-1로 지정하면 컴퓨터의 모든 코어를 사용한다.

* learning\_rate [ 기본값 : 0.3 ]
  + learning rate이다.
  + learning rate가 높을수록 과적합 하기 쉽다.
* n\_estimators [ 기본값 : 100 ]
  + 생성할 weak learner의 수
  + learning\_rate가 낮을 땐, n\_estimators를 높여야 과적합이 방지된다.
* max\_depth [ 기본값 : 6 ]
  + 트리의 maximum depth이다.
  + 적절한 값이 제시되어야 하고 보통 3-10 사이 값이 적용된다.
  + max\_depth가 높을수록 모델의 복잡도가 커져 과적합 하기 쉽다.
* learning rate 낮추기 → n\_estimators은 높여야함
* max\_depth 낮추기
* min\_child\_weight 높이기
* gamma 높이기
* subsample, colsample\_bytree 낮추기

1. **Boosting 이란?**

* 여러 개의 약한 Decision Tree를 조합해서 사용하는 Ensemble 기법 중 하나이다.
  + 즉, 약한 예측 모형들의 학습 에러에 가중치를 두고, 순차적으로 다음 학습 모델에 반영하여
  + 강한 예측모형을 만드는 것이다.

**XGBoost 란?**

* XGBoost는 Extreme Gradient Boosting의 약자이다.
* Boosting 기법을 이용하여 구현한 알고리즘은 Gradient Boost 가 대표적인데
* 이 알고리즘을 병렬 학습이 지원되도록 구현한 라이브러리가 XGBoost 이다.
* Regression, Classification 문제를 모두 지원하며, 성능과 자원 효율이 좋아서, 인기 있게 사용되는 알고리즘이다.

**XGBoost의 장점**

* GBM 대비 빠른 수행시간
  + 병렬 처리로 학습, 분류 속도가 빠르다.
* 과적합 규제(Regularization)
  + 표준 GBM 경우 과적합 규제기능이 없으나, XGBoost는 자체에 과적합 규제 기능으로 강한 내구성 지닌다.
* 분류와 회귀영역에서 뛰어난 예측 성능 발휘
  + 즉, CART(Classification and regression tree) 앙상블 모델을 사용
* Early Stopping(조기 종료) 기능이 있음
* 다양한 옵션을 제공하며 Customizing이 용이하다.