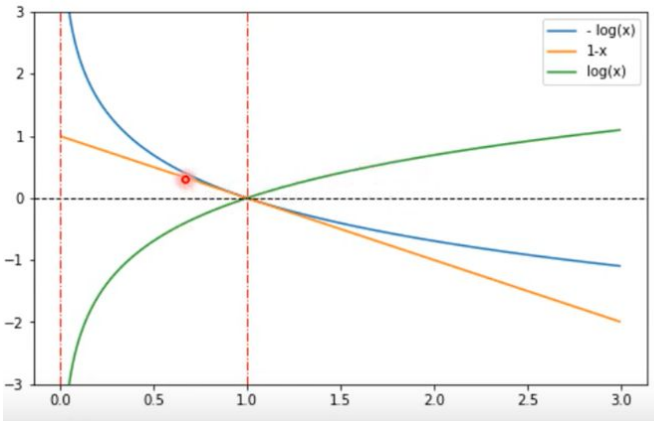
2주차

1-1 log loss 수학적 개념 파악

$$L(\theta, \hat{\theta}) = -\hat{\theta}\log(\theta) - (1-\hat{\theta})\log(1-\theta), \quad \hat{\theta} \in 0, 1, \ \theta \in [0, 1],$$

Log의 역할 (직관적 해석)



확률이 1일 때: -log(1.0) = 0

확률이 0.8일 때: -log(0.8) = 0.22314

확률이 0.6일 때: -log(0.6) = 0.51082

*자연로그 기준 계산결과 입니다

낮을 수록 좋음

1-2 log loss 코드 구현 해보기

1) 날코딩

```
def logloss(true_label, predicted, eps=1e-15):
    p = np.clip(predicted, eps, 1 - eps)
    if true_label == 1:
        return -log(p)
    else:
        return -log(1 - p)
```

2) Scikit-learn

```
from sklearn.metrics import log_loss

log_loss(y_true, y_pred, eps=1e-15)
```

3) Keras

```
model.compile(loss='categorical_crossentropy', optimizer='sgd')
```

1-3 코드로 구현된 평가산식 커스터마이징 (파라미터 수정)

1-4 원자력 대회에서 측정지표를 사용한 이유 탐구

1. log loss 분류 모델 평가시 사용
2. 잘못 예측시 패널티 부가

2-1 이상치 확인 (검색 키워드: Outlier, Boxplot, InterQuartileRange(IQR), 4분위수)

2-2 feature, attribute (5400개) 칼럼 특성 비교 (범주형 vs 수치형) - 독립변수

```
1 train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 600 entries, 0 to 599
Columns: 5122 entries, time to V5120
dtypes: float64(5121), int64(1)
memory usage: 23.4 MB
```

```
time          int64
V0000       float64
V0001       float64
V0002       float64
V0003       float64
              ...
V5116       float64
V5117       float64
V5118       float64
V5119       float64
V5120       float64
Length: 5122, dtype: object
```

2-3 예측해야 할 target(187개): 어떤 데이터 형일까요? - 종속변수

```
1 test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 60 entries, 0 to 59
Columns: 5122 entries, time to V5120
dtypes: float64(5121), int64(1)
memory usage: 2.3 MB
```

```
time          int64
V0000       float64
V0001       float64
V0002       float64
V0003       float64
              ...
V5116       float64
V5117       float64
V5118       float64
V5119       float64
V5120       float64
Length: 5122, dtype: object
```

2-4 독립변수, 종속변수에 따른 상관관계 분석 방법 [통계학적]
Pearson 상관 계수 - 연속형 & 연속형 (모수적 : 변수가 정규성을 따름)

```
corr = df.corr(method = 'pearson')
```

Spearman 상관계수 - 연속형 & 연속형 (비모수적 : 변수가 정규성을 따르지X)

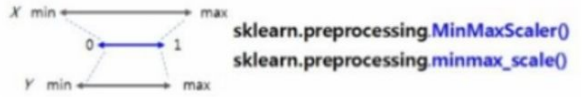
```
corr = df.corr(method = 'Spearman')
```

나머지 잡다 한것들 : https://mansoostat.tistory.com/115

2-5 스케일링(Min-Max 등), 정규화

- **Principal Component Analysis(PCA)**
  - Feature 간에 correlation 결과, high correlation feature 들이 많이 존재하였음
  - Feature간 correlation 이 높은 것들을 선별하여 feature 삭제

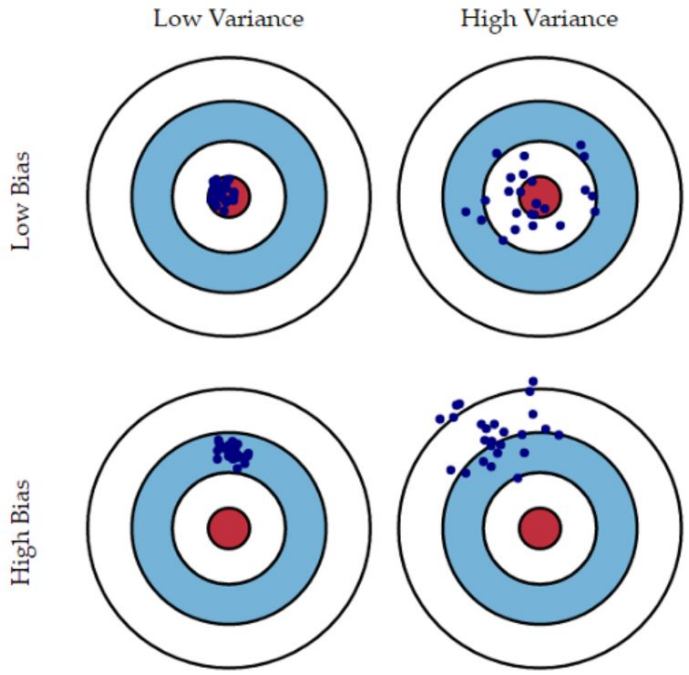
- **표준화 Min-Max Scaling**
  - 데이터가 가진 feature의 스케일이 차이가 나는 경우 Feature 를 정규화 함 (데이터의 중요도를 동일하게 반영되도록 하기 위함)



2-6 전처리 시켜주는 코드 구현

3-1 다른 수상작 우승 코드 학습 (feat.머신러닝 기법 비교)

비교	Bagging	Boosting
특징	병렬 앙상블 모델 (각 모델은 서로 독립적)	연속 앙상블 (이전 모델의 오류를 고려)
목적	Variance 감소	Bias 감소
적합한 상황	복잡한 모델 (High variance, Low bias)	Low variance, High bias 모델
대표 알고리즘	Random Forest	Gradient Boosting, AdaBoost
Sampling	Random Sampling	Random Sampling with weight on error



Boosting 알고리즘

알고리즘	특징	비고
AdaBoost	• 다수결을 통한 정답 분류 및 오답에 가중치 부여	
GBM	• Loss Function의 gradient를 통해 오답에 가중치 부여	gradient_boosting.pdf
Xgboost	• GBM 대비 성능향상 • 시스템 자원 효율적 활용 (CPU, Mem) • Kaggle을 통한 성능 검증 (많은 상위 랭커가 사용)	2014년 공개 boosting-algorithm-xgboost
Light GBM	• Xgboost 대비 성능향상 및 자원소모 최소화 • Xgboost가 처리하지 못하는 대용량 데이터 학습 가능 • Approximates the split (근사치의 분할)을 통한 성능 향상	2016년 공개 light-gbm-vs-xgboost

보통 성능 : Xgboost < LGBM 이지만 결국엔 Random Forest, Xgboost, Light GBM 모델 별로

돌려봐서 성능이 제일 좋은 모델을 선택해야 됨
https://www.slideshare.net/freepsw/boosting-bagging-vs-boosting

422563	submission_1st.csv	2020-04-01 15:03:47	5.2882670307
421520	submission100.csv	2020-03-28 00:09:14	5.2882670307
421519	submission300.csv	2020-03-28 00:08:32	5.2882670307
421518	submission300_538.csv	2020-03-28 00:07:32	5.2882670307
421338	submission10.csv	2020-03-27 01:39:40	1.8068555152
421324	submission1.csv	2020-03-27 00:47:41	2.6131957416
421318	submission.csv	2020-03-27 00:25:42	3.1037470194