1. 머신러닝

머신러닝	머신(컴퓨터)이 데이터르 보고 학습을 하여 점차 지능적인 동작을 하는 것 '모델'을 만드는 것이 목적 ★기본동작 : 데이터 -〉 머신러닝모델 〈-순환-〉 원하는결과
★모델	모델 구조 : 모델 동작의 기본 동작을 구현하는 방법 모델 파라미터 : 모델이 잘 동작하도록 학습한 가중치 등 계수
★훈련	모델이 데이터를 이용하여 모델 파라미터를 학습하는 과정
★검증	모델이 제대로 동작하는지를 검증할 때는 훈련 과정에서 사용하지 않은 새로운 검증 데이터를 사용해야 한다.
과대적합 (overfitting)	모델이 훈련데이터에 대해서만 잘 동작하도록 훈련되어 새로운 데이터에 대해서는 오히려 잘 동작하지 못하는 경우 1)훈련 데이터가 너무 적어서 모델이 학습을 충분히 할 수 없는 경우 2)모델이 너무 복잡하여
일반화	머신러닝에서 과대적합을 피해서 일반적으로 잘 동작하는 모델을 만드는 것
규제화 (regularization)	데이터가 부족하다면 모델 구조를 좀 단순하게 기능을 제약을 가하는 것 릿지, 라쏘, 엘라스틱
과소적합 (underfitting)	모델이 너무 간단하여 성능이 미흡한 경우 섬세한 모델을 만들거나 제약을 풀어 주면 해결, 좀 더 복잡한 모델이 필요
모델 구축	지도학습 : 정답(타켓,라벨)이 있는 상태 비지도 학습 : 정답이 없는 상태 일반적인 과정 1)모델 선택 : 머신러닝 구조를 선택 2)모델 학습 : 훈련 데이터를 이용하여 최적의 모델 파라미터를 구한다. 3)모델 검증 : 과대적합이나 과소적합을 검증하고 최적의 모델 하이퍼파라미터를 선택 4)모델 평가 : 테스트 데이터에 모델을 적용(predict)하고 성능을 평가(score)
배치(batch)	훈련데이터를 일정크기의 배치 단위로 나누어 학습하는 것
이포크(epoch)	주어진 훈련 데이터 전체를 한번 사용하는 것
★경사하강법	최적화기 중 하나. 손실함수를 파라미터에 관한 함수로 나타냈을 때 손실함수가 최솟값으로 빨리 도달하기 위해 현재 파라미터 값을 기울기에 비례하여 반대반향으로 업데이트 하는 방식 크게 배치GD, 확률적GD 두 가지가 있지만 일반적으로 배치GD방식을 주로 사용함
★성능지표	손실함수와 성능지표를 구분할 수 있어야 된다. 회귀에선 * 손실함수 : MSE * 성증지표 : R_sqaure
지도 학습	모델이 정답(레이블, 목적변수, 타겟)을 예측하도록 학습하는 것 회귀 : 예측할 때 분류 : 주어진 샘플이 어느 그룹에 속하는지 판별하는 기능
비지도 학습	정답이 없이 학습하는 유형 군집화, 연관분석, 시각화, 데이터변환, 차원 축소

2.kNN

kNN	거리가 가장 가까운 이웃을 k개 선택하고 이들 레이블의 평균치로 이 샘플이 속할 카테고리를 예측 협업 필터링이라고도 부름
kNN동작	k값을 너무 작게 잡으면 예민 크게잡으면 편균치를 사용하므로 분류가 뮤뎌짐

kNN예제	clf = SGDClassifier() # 선형분리기 불러오기
	knn = KNeighborsClassifier(n_neighbors=3) # k를 3으로 설정
	score(X_test,y_test) # ★입력인자

★교차검증	K-fold : 전체 데이터를 k개로 나누고 성능을 평가하는 방법 ★sklearn에서는 cross_val_score() 함수 기억하기! from sklearn.model_selection import cross_val_score, KFold cross_val_score(knn,X,y,cv=5).mean().round(4)

스케일링	from sklearn.preprocessing import StandardScaler
	sc = StandardScaler() # 스케일링 함수
	X_sc = sc.fit_transform(X) #스케일링 된 훈련데이터!★

3. 결정트리

★순도	지니계수 : 낮은 값이 순도가 높다. 0이 최고야 엔트로피 : 어떤 사건의 "정보량의 기대치" p=0.5일 때 가장 크다. 순도가 높으면 0!!
	max_depth = 결정트리 2층, random_state=7 특성 7개 사용
결정트리 특징	스케일링이 필요없다.

4. 랜덤포레스트

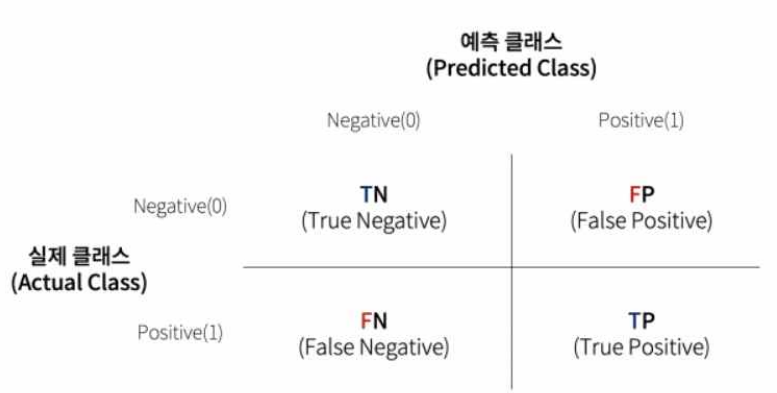
랜덤포레스트	간단한 구조의 결정 트리들을 수십 수백 개를 랜덤하게 만들고 각 결정 트리의 동작 결과를 종합하여 판단하는 방법
앙상블	여러 개의 모델을 만들고 평균을 구하는 방법 직접투표와 간접투표
배깅	부족한 훈련 데이터를 효과적으로 늘리는 방법

5. 서포트 벡터머신

svm	분류 시에 결정 경계를 가능한 넓게, 마진을 갖도록 만든 방법 ★서술

6.분류성능

분류 손실함수	크로스 엔트로피
분류 성능지표	정확도(accuracy) : (TP+TN) / N 정밀도(precesion) : TP / (TP+FP) 재현률(recall) : TP / (TP+FN) F1 : 2 * precension * reall / (precision + recall)
ROC곡선	auc가 클수록 좋다.
혼돈 매트릭스	confusion_matrix(y_test, y_pred)



소프트맥스	다중 값 가장 높은 확률 값을 갖는 것. 모든 결과의 합은 1

7. 특성공학

차원축소	selectPercentile()
주성분 분석	pca = PCA(n_components=2) pca_result = pca.fit_transform(x_all) 차원축소보다 주성분분석 성능이 더 좋다

8.모델 최적화

최적의 머신러닝 모델을 만드는 과정, 과대적합과 과소적합을 피하기 위해

릿지 규제	자승, 큰값을 먼저 줄인다
라쏘 규제	절대값, 작은 값을 먼저 줄인다.
과대적합 검증	훈련데이터와 검증데이터를 비교하여 검증
편향과 분산	편향과 분산 구분할 수 있어야함 편향 : 모델자체가 부정확하여 피할 수 없이 발생하는 오차 분산 : 모델이 너무 복잡하거나 학습데이터에 민감하게 반응하여 예측 이 산발 고편향저분산 =〉 과소적합 저편향고분산 =〉 과대적합

```
lreg.fit(x_train,y_train)
pred_test = lreg.predict(x_test)
mse = np.mean((pred_test - y_test)**2)
print(mse**0.5)
print(lreg.score(x_test,y_test))
```