Scikit-learn 치트시트

1. Scikit-learn 기본 워크플로우

1. **데이터 준비**: 특성(X)과 타겟(y) 분리.
2. **모델 선택**: 문제에 맞는 모델 클래스 임포트.
3. **모델 인스턴스화**: 하이퍼파라미터 설정.
4. **학습**: `model.fit(X, y)`.
5. **예측**: `model.predict(X_new)`.
6. **평가**: 모델 성능 측정.

2. 데이터 전처리 (Preprocessing)

`sklearn.preprocessing` 모듈.

- 스케일링 (Scaling):
  - `StandardScaler`: 평균 0, 분산 1로 조정.
  - `MinMaxScaler`: 0과 1 사이로 조정.
  - `RobustScaler`: 중앙값과 사분위수를 사용하여 이상치에 강함.
- 인코딩 (Encoding):
  - `LabelEncoder`: 범주형 텍스트를 숫자로 변환.
  - `OneHotEncoder`: 범주형 변수를 원-핫 벡터로 변환.
- 결측치 처리:
  - `SimpleImputer`: 평균, 중앙값, 최빈값 등으로 결측치 대체.

```
from sklearn.preprocessing import
StandardScaler
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

3. 모델 선택

지도 학습 (Supervised Learning)

- 회귀 (Regression):
  - `LinearRegression`
  - `Ridge`, `Lasso` (규제가 있는 선형 회귀)
  - `SVR` (Support Vector Regressor)
  - `RandomForestRegressor`
  - `GradientBoostingRegressor`
- 분류 (Classification):
  - `LogisticRegression`
  - `SVC` (Support Vector Classifier)
  - `KNeighborsClassifier`
  - `DecisionTreeClassifier`
  - `RandomForestClassifier`
  - `GradientBoostingClassifier`

비지도 학습 (Unsupervised Learning)

- 군집 (Clustering):
  - `KMeans`
  - `DBSCAN`
  - `AgglomerativeClustering`
- 차원 축소 (Dimensionality Reduction):
  - `PCA` (Principal Component Analysis)
  - `TSNE` (t-Distributed Stochastic Neighbor Embedding)

4. 데이터 분할 및 교차 검증

`sklearn.model_selection` 모듈.

- 학습/테스트 데이터 분할:

```
from sklearn.model_selection import
train_test_split
X_train, X_test, y_train, y_test =
train_test_split(
    X, y, test_size=0.2,
random_state=42
)
```

- 교차 검증 (Cross-Validation):

```
from sklearn.model_selection import
cross_val_score
scores = cross_val_score(model, X, y,
cv=5) # 5-fold CV
```

5. 모델 평가 (Metrics)

`sklearn.metrics` 모듈.

회귀 평가 지표

- `mean_absolute_error` (MAE)
- `mean_squared_error` (MSE)
- `r2_score` (R-squared)

분류 평가 지표

- `accuracy_score`: 정확도
- `precision_score`: 정밀도
- `recall_score`: 재현율
- `f1_score`: F1 점수
- `confusion_matrix`: 혼동 행렬
- `roc_auc_score`: ROC 곡선 아래 면적 (AUC)
- `classification_report`: 주요 지표 요약

```
from sklearn.metrics import
accuracy_score, classification_report

y_pred = model.predict(X_test)
print(accuracy_score(y_test, y_pred))
print(classification_report(y_test,
y_pred))
```

6. 하이퍼파라미터 튜닝

- 그리드 서치 (Grid Search):

```
from sklearn.model_selection import
GridSearchCV
param_grid = {'n_estimators': [100,
200], 'max_depth': [3, 5]}
grid_search = GridSearchCV(model,
param_grid, cv=5)
grid_search.fit(X_train, y_train)
print(grid_search.best_params_)
```

- 랜덤 서치 (Randomized Search): `RandomizedSearchCV`

7. 파이프라인 (Pipeline)

전처리 단계와 모델 학습을 하나로 묶어 코드 중복을 줄이고 실수를 방지합니다.

```
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import
StandardScaler
from sklearn.svm import SVC

pipe = Pipeline([
    ('scaler', StandardScaler()),
    ('svc', SVC())
])

pipe.fit(X_train, y_train)
pipe.score(X_test, y_test)
```