위스콘신 유방암 데이터 셋 - LightGBM 모델 구현

학습 목표

- LightGBM 모델을 이용하여 학습과 예측을 수행해 본다.
- 예측을 수행한 결과에 대해 평가를 수행해 본다.

학습 내용

- LightGBM 모델 만들기 - 위스콘신 유방암 데이터 셋
- 평가해보기

데이터 로드 및 전처리

In [15]:

```
import pandas as pd
from sklearn.datasets import load_breast_cancer
import matplotlib.pyplot as plt
import matplotlib

from lightgbm import LGBMClassifier
from sklearn.model_selection import train_test_split
```

In [16]:

```
cancer = load_breast_cancer()

cancer_df = pd.DataFrame(cancer.data, columns=cancer.feature_names)
cancer_df.head()
```

Out[16]:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mear symmetry
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.241
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.181
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.206
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.259
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.180

5 rows × 30 columns

In [17]:

```
print( cancer_df.shape)
```

(569, 30)

데이터 설명

- 위스콘신 유방암 데이터 세트는 유방암의 악성 종양, 양성 종양 여부를 결정하는 이진 분류
- 종양의 크기, 모양 등의 형태와 관련한 많은 피처를 가지고 있음.
- 569개의 행과, 30개의 피처로 이루어진 데이터
- null 값이 없음. 값들은 실수로 되어 있음.

데이터 나누기

In [18]:

```
# 피처와 레이블를 지정.
X = cancer_df[:]
y = cancer.target

X.shape, y.shape
```

Out[18]:

((569, 30), (569,))

In [19]:

```
X_train , X_test, y_train, y_test = train_test_split(X, y,
                                                     test_size=0.2, random_state=0)

X_train.shape, X_test.shape, y_train.shape, y_test.shape
```

Out[19]:

((455, 30), (114, 30), (455,), (114,))

02 LightGBM

- LightGBM은 XGBoost와 부스팅 계열 알고리즘에서 가장 각광을 받고 있음.
- LightGBM은 가장 큰 장점은 XGBoost보다 학습에 걸리는 시간이 훨씬 적다. 메모리 사용량도 적다.
- LightGBM과 XGBoost의 예측 성능은 별다른 차이가 없음.
- LightGBM이 XGBoost보다 2년 후에 만들어짐.
- [단점] 적은 데이터 셋일 경우, 과적합이 발생할 가능성이 있음. (문서상에는 10000건 이하의 데이터로 기술하고 있음.)

- lightgbm 설치

- pip install lightgbm

LightGBM의 파이썬 패키지인 lightgbm에서 LGBMClassifier 불러오기

In [20]:

```
from lightgbm import LGBMClassifier
```

In [21]:

```
# 모델 선택
model_lgbm = LGBMClassifier(n_estimators= 400)

# Lgbm은 중간에 조기 중단이 가능.
evals = [(X_test, y_test)]

model_lgbm.fit(X_train, y_train, early_stopping_rounds=100,
               eval_metric='logloss',
               eval_set=evals,
               verbose=True)

preds = model_lgbm.predict(X_test)
pred_proba = model_lgbm.predict_proba(X_test)[:, 1]
```

```
[149]   valid_0's binary_logloss: 0.0634037
[150]   valid_0's binary_logloss: 0.0638329
[151]   valid_0's binary_logloss: 0.0636558
[152]   valid_0's binary_logloss: 0.0632649



C:₩Users₩withJesus₩anaconda3₩lib₩site-packages₩lightgbm₩sklearn.py:726: UserWarni
ng: 'early_stopping_rounds' argument is deprecated and will be removed in a futur
e release of LightGBM. Pass 'early_stopping()' callback via 'callbacks' argument
instead.
  _log_warning("'early_stopping_rounds' argument is deprecated and will be remove
d in a future release of LightGBM. "
C:₩Users₩withJesus₩anaconda3₩lib₩site-packages₩lightgbm₩sklearn.py:736: UserWarni
ng: 'verbose' argument is deprecated and will be removed in a future release of L
ightGBM. Pass 'log_evaluation()' callback via 'callbacks' argument instead.
  _log_warning("'verbose' argument is deprecated and will be removed in a future
release of LightGBM. "
```

- 조기 중단으로 400번까지 수행하지 않고, 중간에 중단함.

In [49]:

```
from sklearn import metrics
```

In [58]:

```
# 모델 평가를 위한 함수 설정
def get_clf_eval(y_test, y_pred=None, pred_proba=None):
  confusion = metrics.confusion_matrix(y_test, y_pred)
  accuracy = metrics.accuracy_score(y_test, y_pred)
  precision = metrics.precision_score(y_test, y_pred)
  recall = metrics.recall_score(y_test, y_pred)
  F1_score = metrics.f1_score(y_test, y_pred)

  AUC = metrics.roc_auc_score(y_test, pred_proba)

  # 평가지표 출력
  print('오차행렬:₩n', confusion)
  print('₩n정확도: {:.4f}'.format(accuracy))
  print('정밀도: {:.4f}'.format(precision))
  print('재현율: {:.4f}'.format(recall))
  print('F1: {:.4f}'.format(F1_score))
  print('AUC: {:.4f}'.format(AUC))
```

In [59]:

```
get_clf_eval(y_test, preds, pred_proba)
```

```
오차행렬:
 [[44  3]
 [ 1 66]]

정확도: 0.9649
정밀도: 0.9565
재현율: 0.9851
F1: 0.9706
AUC: 0.9990
```

피처 중요도 시각화

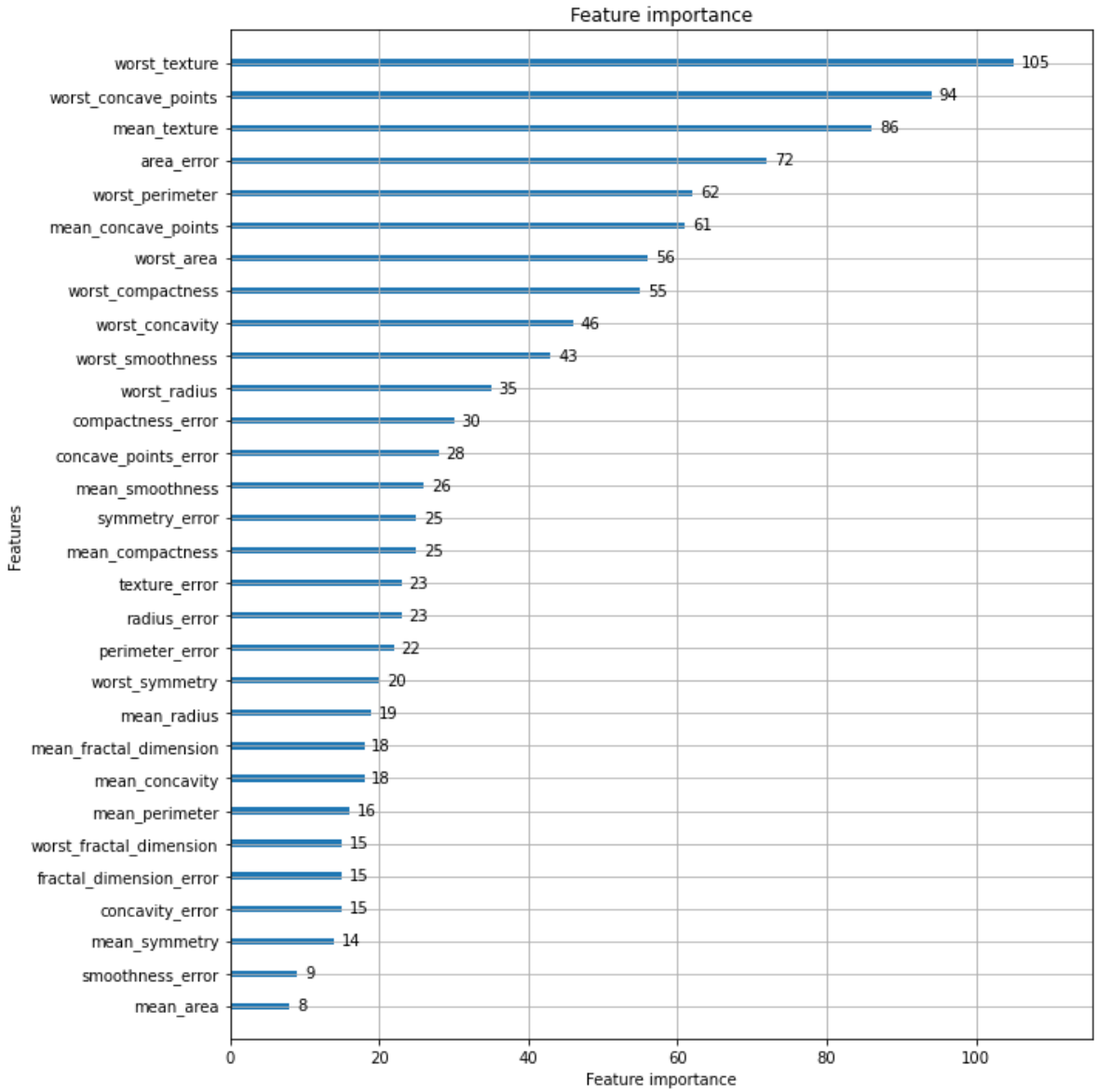
In [61]:

```
from lightgbm import plot_importance
import matplotlib.pyplot as plt

fig, ax = plt.subplots(figsize=(10,12))
plot_importance(model_lgbm, ax=ax)
```

Out[61]:

<AxesSubplot:title={'center':'Feature importance'}, xlabel='Feature importance', ylabel='Features'>



정리

- lightGBM은 가장 많이 쓰이는 빠르고 성능 좋은 머신러닝 알고리즘 중의 하나이다.