Breast Cancer Prediction

In [1]:
```
import numpy as np
import pandas as pd
import sklearn
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.model_selection import KFold
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import cross_val_score
from sklearn.preprocessing import LabelEncoder
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.datasets import load_breast_cancer
from sklearn import metrics
%matplotlib inline

import os
import warnings
warnings.filterwarnings('ignore')
```

In [10]:
```
vicky =pd.read_csv(r"C:\Users\Vicky Yewle\Downloads\Machine Learning\Datasets\BreastCancerPrediction\Breast Cancer Prediction\Data
vicky.head()
```

Out[10]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	...
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	...
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	...
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	...
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	...

5 rows × 32 columns

In [11]:
```
vicky.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 32 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   id                       569 non-null    int64  
 1   diagnosis                569 non-null    object 
 2   radius_mean              569 non-null    float64
 3   texture_mean             569 non-null    float64
 4   perimeter_mean           569 non-null    float64
 5   area_mean                569 non-null    float64
 6   smoothness_mean          569 non-null    float64
 7   compactness_mean         569 non-null    float64
 8   concavity_mean           569 non-null    float64
 9   concave points_mean      569 non-null    float64
 10  symmetry_mean            569 non-null    float64
 11  fractal_dimension_mean   569 non-null    float64
 12  radius_se                569 non-null    float64
 13  texture_se               569 non-null    float64
 14  perimeter_se             569 non-null    float64
 15  area_se                  569 non-null    float64
 16  smoothness_se            569 non-null    float64
 17  compactness_se           569 non-null    float64
 18  concavity_se             569 non-null    float64
 19  concave points_se        569 non-null    float64
 20  symmetry_se              569 non-null    float64
 21  fractal_dimension_se     569 non-null    float64
 22  radius_worst             569 non-null    float64
 23  texture_worst            569 non-null    float64
 24  perimeter_worst          569 non-null    float64
 25  area_worst               569 non-null    float64
 26  smoothness_worst         569 non-null    float64
 27  compactness_worst        569 non-null    float64
 28  concavity_worst          569 non-null    float64
 29  concave points_worst     569 non-null    float64
 30  symmetry_worst           569 non-null    float64
 31  fractal_dimension_worst  569 non-null    float64
dtypes: float64(30), int64(1), object(1)
memory usage: 142.4+ KB
```

In [12]:
```
del vicky['id']
```

In [13]:
```
vicky.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 31 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   diagnosis                569 non-null    object 
 1   radius_mean              569 non-null    float64
 2   texture_mean             569 non-null    float64
 3   perimeter_mean           569 non-null    float64
 4   area_mean                569 non-null    float64
 5   smoothness_mean          569 non-null    float64
 6   compactness_mean         569 non-null    float64
 7   concavity_mean           569 non-null    float64
 8   concave points_mean      569 non-null    float64
 9   symmetry_mean            569 non-null    float64
 10  fractal_dimension_mean   569 non-null    float64
 11  radius_se                569 non-null    float64
 12  texture_se               569 non-null    float64
 13  perimeter_se             569 non-null    float64
 14  area_se                  569 non-null    float64
 15  smoothness_se            569 non-null    float64
 16  compactness_se           569 non-null    float64
 17  concavity_se             569 non-null    float64
 18  concave points_se        569 non-null    float64
 19  symmetry_se              569 non-null    float64
 20  fractal_dimension_se     569 non-null    float64
 21  radius_worst             569 non-null    float64
 22  texture_worst            569 non-null    float64
 23  perimeter_worst          569 non-null    float64
 24  area_worst               569 non-null    float64
 25  smoothness_worst         569 non-null    float64
 26  compactness_worst        569 non-null    float64
 27  concavity_worst          569 non-null    float64
 28  concave points_worst     569 non-null    float64
 29  symmetry_worst           569 non-null    float64
 30  fractal_dimension_worst  569 non-null    float64
dtypes: float64(30), object(1)
memory usage: 137.9+ KB
```

In [14]:

```
vicky['target']=np.where(vicky['diagnosis']=='B',0,1)
del vicky['diagnosis']
vicky.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 31 columns):
 #   Column                   Non-Null Count  Dtype  
```

```
---  ------                  --------------  -----
 0   radius_mean             569 non-null    float64
 1   texture_mean            569 non-null    float64
 2   perimeter_mean          569 non-null    float64
 3   area_mean               569 non-null    float64
 4   smoothness_mean         569 non-null    float64
 5   compactness_mean        569 non-null    float64
 6   concavity_mean          569 non-null    float64
 7   concave points_mean     569 non-null    float64
 8   symmetry_mean           569 non-null    float64
 9   fractal_dimension_mean  569 non-null    float64
 10  radius_se               569 non-null    float64
 11  texture_se              569 non-null    float64
 12  perimeter_se            569 non-null    float64
 13  area_se                 569 non-null    float64
 14  smoothness_se           569 non-null    float64
 15  compactness_se          569 non-null    float64
 16  concavity_se            569 non-null    float64
 17  concave points_se       569 non-null    float64
 18  symmetry_se             569 non-null    float64
 19  fractal_dimension_se    569 non-null    float64
 20  radius_worst            569 non-null    float64
 21  texture_worst           569 non-null    float64
 22  perimeter_worst         569 non-null    float64
 23  area_worst              569 non-null    float64
 24  smoothness_worst        569 non-null    float64
 25  compactness_worst       569 non-null    float64
 26  concavity_worst         569 non-null    float64
 27  concave points_worst    569 non-null    float64
 28  symmetry_worst          569 non-null    float64
 29  fractal_dimension_worst 569 non-null    float64
 30  target                  569 non-null    int32
dtypes: float64(30), int32(1)
memory usage: 135.7 KB
```

In [15]:
```
vicky.target.value_counts()
```

Out[15]:
```
0    357
1    212
Name: target, dtype: int64
```

In [16]:
```
X= vicky.drop('target',axis=1)
y = vicky[['target']]

X_train,X_test, y_train, y_test = train_test_split(X,y, test_size = 0.20, random_state=100)
```

```
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(455, 30)
(114, 30)
(455, 1)
(114, 1)
```

In [17]:
```
print(y_train.mean())
print(y_test.mean())
```

```
target    0.358242
dtype: float64
target    0.429825
dtype: float64
```

In [18]:
```
# shallow tree is tree having small depth

shallow_tree = DecisionTreeClassifier(max_depth=2, random_state= 100)
```

In [21]:
```
#Random_state used for variable notations it should be same for all group if not it will change variation

shallow_tree.fit(X_train, y_train)

y_pred = shallow_tree.predict(X_test)
score = metrics.accuracy_score(y_test, y_pred)
score
```

Out[21]:
```
0.9649122807017544
```

In [22]:
```
param_grid = {"base_estimator__max_depth":[2,5],
              "n_estimators": [200,400,600,700]
              }
```

In [23]:
```
tree = DecisionTreeClassifier()

ABC = AdaBoostClassifier(base_estimator=tree, learning_rate=0.6,algorithm="SAMME")
```

In [24]:
```
folds = 3

grid_search_ABC = GridSearchCV(ABC,
                               cv=folds,
```

```
                                param_grid = param_grid,
                                scoring= 'roc_auc',
                                return_train_score=True,
                                verbose=1)
```

In [25]:
```
grid_search_ABC.fit(X_train,y_train)
```

```
Fitting 3 folds for each of 8 candidates, totalling 24 fits
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  24 out of  24 | elapsed:   27.9s finished
```

Out[25]:
```
GridSearchCV(cv=3,
             estimator=AdaBoostClassifier(algorithm='SAMME',
                                          base_estimator=DecisionTreeClassifier(),
                                          learning_rate=0.6),
             param_grid={'base_estimator__max_depth': [2, 5],
                         'n_estimators': [200, 400, 600, 700]},
             return_train_score=True, scoring='roc_auc', verbose=1)
```

In [26]:
```
ABC = grid_search_ABC.best_estimator_
```

In [27]:
```
ABC.fit(X_train,y_train)
```

Out[27]:
```
AdaBoostClassifier(algorithm='SAMME',
                   base_estimator=DecisionTreeClassifier(max_depth=2),
                   learning_rate=0.6, n_estimators=600)
```

In [28]:
```
predictions_train = ABC.predict(X_train)
```

In [29]:
```
predictions=ABC.predict(X_test)
```

In [30]:
```
from sklearn.metrics import r2_score
print(r2_score(y_true=y_train,y_pred=predictions_train))

from sklearn.metrics import r2_score
r2_score(y_true=y_test,y_pred=predictions)
```

```
1.0
```

Out[30]:
```
0.928414442700157
```

In [31]:
```
predictions
```

Out[31]:
```
array([1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0,
       1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1,
```

```
0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0,
1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0,
1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,
1, 0, 1, 0])
```

In [32]:
```
import xgboost as xgb
from xgboost import XGBClassifier
from xgboost import plot_importance
import gc
%matplotlib
```

```
Using matplotlib backend: Qt5Agg
```

In [33]:
```
# roc_auc


folds=3

param_grid = {'learning_rate':[0.2,0.6],
              'subsample':[0.3, 0.6, 0.9]}

xgb_model = XGBClassifier(max_depth=2, n_estimators=200)

model_cv = GridSearchCV(estimator= xgb_model,
                        param_grid= param_grid,
                        scoring='roc_auc',
                        cv= folds,
                        verbose= 1,
                        return_train_score= True)
```

In [34]:
```
model_cv.fit(X_train,y_train)
```

```
Fitting 3 folds for each of 6 candidates, totalling 18 fits
[20:54:07] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.3.0/src/learner.cc:1061: Starting in XGBoost 1.3.0, t
he default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_
metric if you'd like to restore the old behavior.
[20:54:07] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.3.0/src/learner.cc:1061: Starting in XGBoost 1.3.0, t
he default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_
metric if you'd like to restore the old behavior.
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[20:54:08] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.3.0/src/learner.cc:1061: Starting in XGBoost 1.3.0, t
he default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_
metric if you'd like to restore the old behavior.
[20:54:08] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.3.0/src/learner.cc:1061: Starting in XGBoost 1.3.0, t
he default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_
```

```
metric if you'd like to restore the old behavior.
[20:54:08] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.3.0/src/learner.cc:1061: Starting in XGBoost 1.3.0, t
he default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_
metric if you'd like to restore the old behavior.
[20:54:08] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.3.0/src/learner.cc:1061: Starting in XGBoost 1.3.0, t
he default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_
metric if you'd like to restore the old behavior.
[20:54:08] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.3.0/src/learner.cc:1061: Starting in XGBoost 1.3.0, t
he default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_
metric if you'd like to restore the old behavior.
[20:54:08] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.3.0/src/learner.cc:1061: Starting in XGBoost 1.3.0, t
he default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_
metric if you'd like to restore the old behavior.
[20:54:08] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.3.0/src/learner.cc:1061: Starting in XGBoost 1.3.0, t
he default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_
metric if you'd like to restore the old behavior.
[20:54:08] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.3.0/src/learner.cc:1061: Starting in XGBoost 1.3.0, t
he default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_
metric if you'd like to restore the old behavior.
[20:54:08] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.3.0/src/learner.cc:1061: Starting in XGBoost 1.3.0, t
he default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_
metric if you'd like to restore the old behavior.
[20:54:09] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.3.0/src/learner.cc:1061: Starting in XGBoost 1.3.0, t
he default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_
metric if you'd like to restore the old behavior.
[20:54:09] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.3.0/src/learner.cc:1061: Starting in XGBoost 1.3.0, t
he default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_
metric if you'd like to restore the old behavior.
[20:54:09] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.3.0/src/learner.cc:1061: Starting in XGBoost 1.3.0, t
he default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_
metric if you'd like to restore the old behavior.
[20:54:09] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.3.0/src/learner.cc:1061: Starting in XGBoost 1.3.0, t
he default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_
metric if you'd like to restore the old behavior.
[20:54:09] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.3.0/src/learner.cc:1061: Starting in XGBoost 1.3.0, t
he default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_
metric if you'd like to restore the old behavior.
[20:54:09] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.3.0/src/learner.cc:1061: Starting in XGBoost 1.3.0, t
he default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_
metric if you'd like to restore the old behavior.
[20:54:09] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.3.0/src/learner.cc:1061: Starting in XGBoost 1.3.0, t
he default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_
metric if you'd like to restore the old behavior.
[Parallel(n_jobs=1)]: Done  18 out of  18 | elapsed:    2.1s finished
[20:54:09] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.3.0/src/learner.cc:1061: Starting in XGBoost 1.3.0, t
he default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_
metric if you'd like to restore the old behavior.
```

Out[34]:
```
GridSearchCV(cv=3,
             estimator=XGBClassifier(base_score=None, booster=None,
                                     colsample_bylevel=None,
                                     colsample_bynode=None,
                                     colsample_bytree=None, gamma=None,
                                     gpu_id=None, importance_type='gain',
                                     interaction_constraints=None,
                                     learning_rate=None, max_delta_step=None,
                                     max_depth=2, min_child_weight=None,
                                     missing=nan, monotone_constraints=None,
                                     n_estimators=200, n_jobs=None,
                                     num_parallel_tree=None, random_state=None,
                                     reg_alpha=None, reg_lambda=None,
                                     scale_pos_weight=None, subsample=None,
                                     tree_method=None, validate_parameters=None,
                                     verbosity=None),
             param_grid={'learning_rate': [0.2, 0.6],
                         'subsample': [0.3, 0.6, 0.9]},
             return_train_score=True, scoring='roc_auc', verbose=1)
```

In [35]:
```
cv_results = pd.DataFrame(model_cv.cv_results_)
cv_results
```

Out[35]:

	mean_fit_time	std_fit_time	mean_score_time	std_score_time	param_learning_rate	param_subsample	params	split0_test_score	split1_test_score
0	0.102006	0.010152	0.007877	0.006383	0.2	0.3	{'learning_rate': 0.2, 'subsample': 0.3}	0.994331	0.992877
1	0.096786	0.002438	0.005209	0.007366	0.2	0.6	{'learning_rate': 0.2, 'subsample': 0.6}	0.997732	0.992690
2	0.110450	0.004586	0.008000	0.000002	0.2	0.9	{'learning_rate': 0.2, 'subsample': 0.9}	0.996977	0.994377
3	0.084713	0.002023	0.002667	0.003771	0.6	0.3	{'learning_rate': 0.6, 'subsample': 0.3}	0.995654	0.989128

	mean_fit_time	std_fit_time	mean_score_time	std_score_time	param_learning_rate	param_subsample	params	split0_test_score	split1_test_score
4	0.091689	0.013630	0.010418	0.007366	0.6	0.6	{'learning_rate': 0.6, 'subsample': 0.6}	0.991497	0.994189
5	0.120097	0.044122	0.026616	0.021163	0.6	0.9	{'learning_rate': 0.6, 'subsample': 0.9}	0.994142	0.992690

In [36]:
```
Xgb = model_cv.best_estimator_
```

In [37]:
```
Xgb.fit(X_train,y_train)
```

```
[20:54:10] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.3.0/src/learner.cc:1061: Starting in XGBoost 1.3.0, t
he default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_
metric if you'd like to restore the old behavior.
```

Out[37]:
```
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=1, gamma=0, gpu_id=-1,
              importance_type='gain', interaction_constraints='',
              learning_rate=0.2, max_delta_step=0, max_depth=2,
              min_child_weight=1, missing=nan, monotone_constraints='()',
              n_estimators=200, n_jobs=8, num_parallel_tree=1, random_state=0,
              reg_alpha=0, reg_lambda=1, scale_pos_weight=1, subsample=0.6,
              tree_method='exact', validate_parameters=1, verbosity=None)
```

In [38]:
```
predictions_train = Xgb.predict(X_train)
```

In [39]:
```
predictions=Xgb.predict(X_test)
```

In [40]:
```
from sklearn.metrics import r2_score
print(r2_score(y_true=y_train,y_pred=predictions_train))

from sklearn.metrics import r2_score
print(r2_score(y_true=y_test,y_pred=predictions))
```

```
1.0
0.928414442700157
```