In [1]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn.model_selection import train_test_split, RandomizedSearchCV
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier
from sklearn.svm import SVC


%matplotlib inline
```

In [2]:

```
Data = pd.read_csv(r"C:\Users\pandarinath\OneDrive\Desktop\Data\diabetes (1).csv")
```

In [3]:

```
Data.shape
```

Out[3]:

```
(768, 9)
```

In [4]:

```
Data.head()
```

Out[4]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

In [5]:

```
Data.isnull().values.any()
```

Out[5]:

```
False
```

In [6]:

```
Data.describe()
```

Out[6]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000

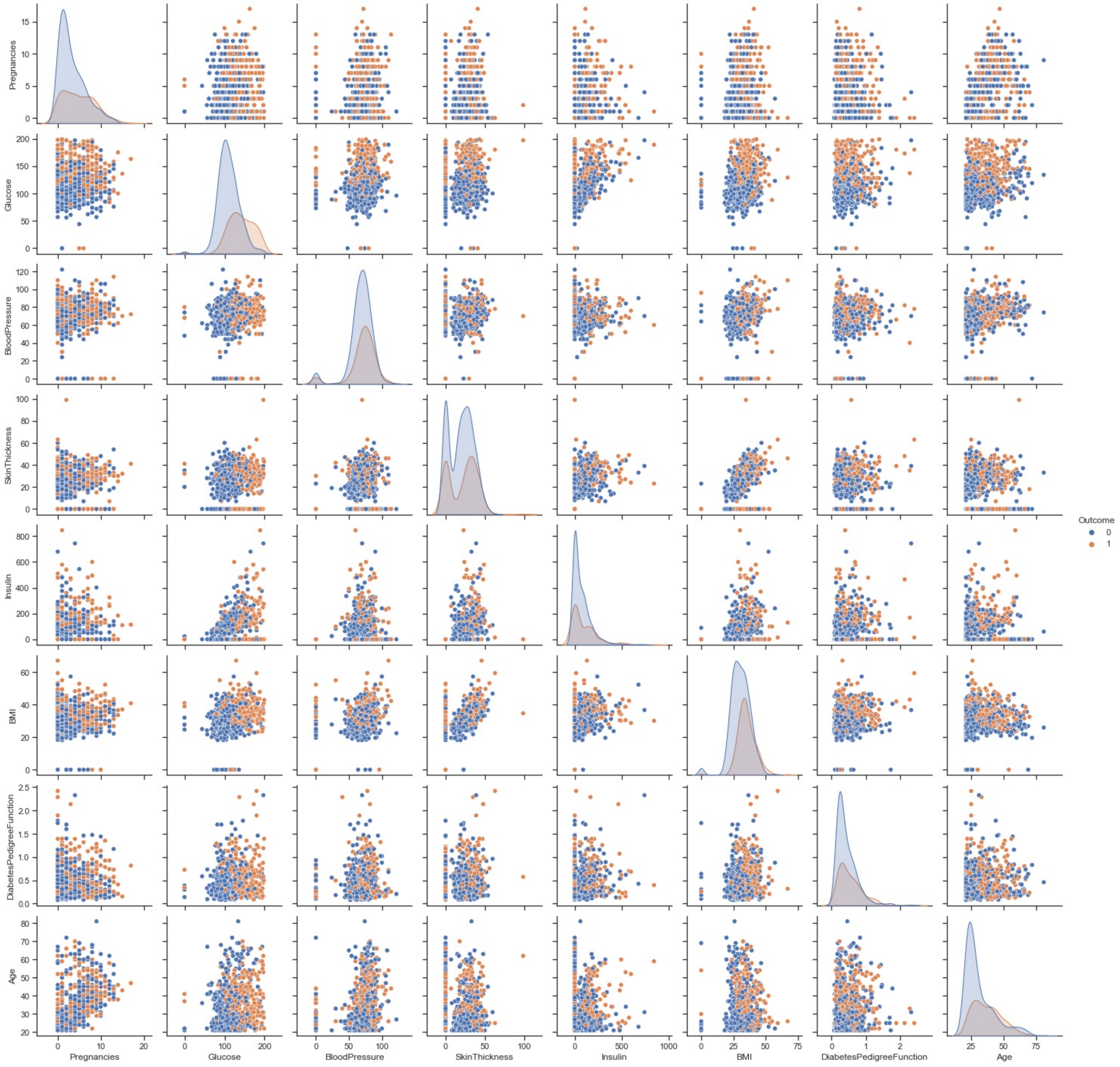
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000

In [7]:

```
import seaborn as sns
sns.set(style="ticks")
```

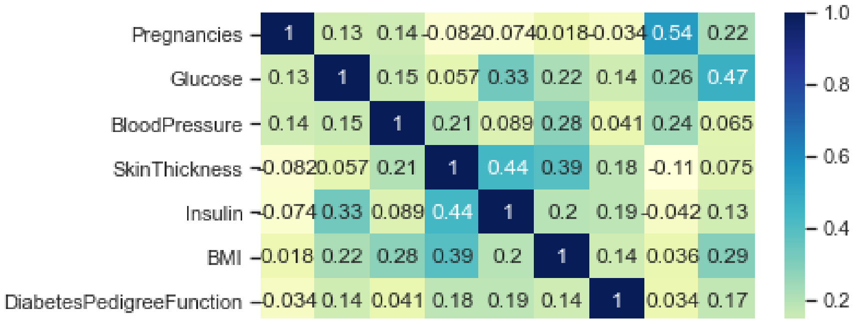
In [11]:

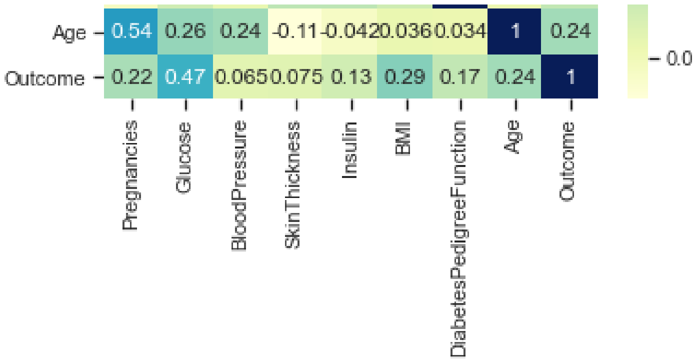
```
sns.pairplot(Data, hue="Outcome");
```



In [12]:

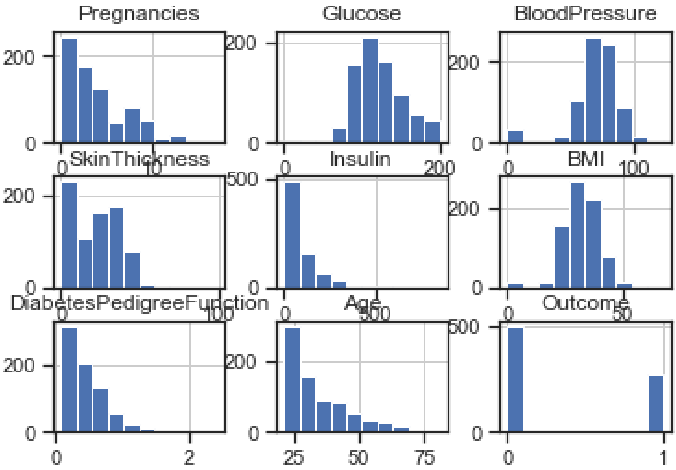
```
sns.heatmap(Data.corr(), annot=True, cmap="YlGnBu");
```





In [13]:

```
Data.hist();
```



In [14]:

```
a = '0.65'
b = '0'
c = 'Age'
d = '0.35'
e = 'Glucose'
f = '0.5'
g = "More than zero"
answers_one = {
    'The proportion of diabetes outcomes in the dataset': d,
    'The number of missing data points in the dataset': b,
    'A dataset with a symmetric distribution': e,
    'A dataset with a right-skewed distribution': c,
    'This variable has the strongest correlation with the outcome': e
}
```

In [15]:

```
Outcome_true_count =len(Data.loc[Data['Outcome']==1])
Outcome_false_count =len(Data.loc[Data['Outcome']==0])
```

In [16]:

```
from sklearn.model_selection import train_test_split
feature_columns = ['Pregnancies','Glucose','BloodPressure','SkinThickness','Insulin','BMI
','DiabetesPedigreeFunction','Age']
predicted_class = ['Outcome']
```

In [17]:

```
X = Data[feature_columns].values
y = Data[predicted_class].values
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.30,random_state=10)
```

In [18]:

```
print("total number of rows : {0}".format(len(Data)))
print("total number of rows : {0}".format(len(Data)))
print("number of rows missing Glucose : {0}".format(len(Data.loc[Data['Glucose']==0])))
print("number of rows missing BloodPressure : {0}".format(len(Data.loc[Data['BloodPressure']==0])))
print("number of rows missing SkinThickness : {0}".format(len(Data.loc[Data['SkinThickness']==0])))
print("number of rows missing Insulin : {0}".format(len(Data.loc[Data['Insulin']==0])))
print("number of rows missing BMI : {0}".format(len(Data.loc[Data['BMI']==0])))
print("number of rows missing DiabetesPedigreeFunction : {0}".format(len(Data.loc[Data['DiabetesPedigreeFunction']==0])))
print("number of rows missing Age : {0}".format(len(Data.loc[Data['Age']==0])))
```

```
total number of rows : 768
number of rows missing Glucose : 5
number of rows missing BloodPressure : 35
number of rows missing SkinThickness : 227
number of rows missing Insulin : 374
number of rows missing BMI : 11
number of rows missing DiabetesPedigreeFunction : 0
number of rows missing Age : 0
```

In [19]:

```
from sklearn.impute import SimpleImputer
fill_values = SimpleImputer(missing_values=0,strategy="mean")
X_train = fill_values.fit_transform(X_train)
X_test = fill_values.fit_transform(X_test)
```

In [20]:

```
from sklearn.ensemble import RandomForestClassifier
random_forest_model = RandomForestClassifier(random_state=10)
random_forest_model.fit(X_train,y_train.ravel())
```

Out[20]:

```
RandomForestClassifier(random_state=10)
```

In [21]:

```
predict_train_Data = random_forest_model.predict(X_test)
from sklearn import metrics
print("Accuracy ={0: .3f}".format(metrics.accuracy_score(y_test,predict_train_Data)))
```

```
Accuracy = 0.766
```

In [22]:

```
params={
    "learning rate"  : [0.05,0.10,0.15,0.20,0.25,0.30],
    "max_depth"      : [3,4,5,6,8,10,12,15],
    "min_child_weight": [1,3,5,7],
    "gamma"          :     [0.0,0.1,0.2,0.3,0.4],
    "colsample_bytree" : [0.3,0.4,0.5,0.7]
}
```

In [23]:

```
from sklearn.model_selection import RandomizedSearchCV
import xgboost
```

In [36]:

```
classifier=xgboost.XGBClassifier()
```

In [38]:

```
random_search=RandomizedSearchCV(classifier,param_distributions=params,n_iter=5,scoring='roc_auc',n_jobs=-1,cv=5,verbose=3)
```

In [39]:

```
def timer(start_time=None):
    if not start_time:
        start_time = datetime.now()
        return start_time
    elif start_time:
        thour, temp_sec = divmod((datetime.now() - start_time).total_seconds(), 3600)
        tmin, tsec = divmod(temp_sec, 60)
        print('\n Time taken: %i hours %i minutes and %s seconds.' % (thour, tmin, round
(tsec, 2)))
```

In [40]:

```
from datetime import datetime
start_time = timer(None)
random_search.fit(X,y.ravel())
timer(start_time)
```

```
Fitting 5 folds for each of 5 candidates, totalling 25 fits
```

```
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 12 concurrent workers.
[Parallel(n_jobs=-1)]: Done  11 out of  25 | elapsed:    7.6s remaining:    9.7s
[Parallel(n_jobs=-1)]: Done  20 out of  25 | elapsed:    8.4s remaining:    2.0s
[Parallel(n_jobs=-1)]: Done  25 out of  25 | elapsed:    8.7s finished
C:\Users\pandarinath\anaconda3\lib\site-packages\xgboost\sklearn.py:888: UserWarning: The
use of label encoder in XGBClassifier is deprecated and will be removed in a future relea
se. To remove this warning, do the following: 1) Pass option use_label_encoder=False when
constructing XGBClassifier object; and 2) Encode your labels (y) as integers starting wit
h 0, i.e. 0, 1, 2, ..., [num_class - 1].
  warnings.warn(label_encoder_deprecation_msg, UserWarning)
```

```
[09:53:02] WARNING: ..\src\learner.cc:541:
Parameters: { learning rate } might not be used.

  This may not be accurate due to some parameters are only used in language bindings but
  passed down to XGBoost core.  Or some parameters are not used but slip through this
  verification. Please open an issue if you find above cases.


[09:53:02] WARNING: ..\src\learner.cc:1061: Starting in XGBoost 1.3.0, the default evalua
tion metric used with the objective 'binary:logistic' was changed from 'error' to 'loglos
s'. Explicitly set eval_metric if you'd like to restore the old behavior.

 Time taken: 0 hours 0 minutes and 9.07 seconds.
```

In [41]:

```
random_search.best_estimator_
```

Out[41]:

```
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=0.4, gamma=0.4, gpu_id=-1,
              importance_type='gain', interaction_constraints='',
              learning rate=0.2, learning_rate=0.300000012, max_delta_step=0,
              max_depth=15, min_child_weight=1, missing=nan,
              monotone_constraints='()', n_estimators=100, n_jobs=12,
              num_parallel_tree=1, random_state=0, reg_alpha=0, reg_lambda=1,
              scale_pos_weight=1, subsample=1, tree_method='exact',
              validate_parameters=1, verbosity=None)
```

In [42]:

```
classifier=xgboost.XGBClassifier(base_score=0.5,booster='gbtree',colsample_bylevel=1,cols
ample_bytree=0.7,gamma=0.1,learning_rate=0.05,max_delta_step=0,max_depth=5,min_child_weig
ht=7,missing=None,n_estimators=100,n_jobs=1,nthread=None,objective='binary:logistic',ran
dom_state=0,reg_alpha=0,reg_lambda=1,scale_pos_weight=1,seed=None,silent=True,subsample=
1)
```

In [43]:

```
classifier.fit(X_train,y_train)
```

```
[09:53:22] WARNING: ..\src\learner.cc:541:
```

```
[09:53:22] WARNING: ..\src\learner.cc:541:
Parameters: { silent } might not be used.

  This may not be accurate due to some parameters are only used in language bindings but
  passed down to XGBoost core.  Or some parameters are not used but slip through this
  verification. Please open an issue if you find above cases.


[09:53:22] WARNING: ..\src\learner.cc:1061: Starting in XGBoost 1.3.0, the default evalua
tion metric used with the objective 'binary:logistic' was changed from 'error' to 'loglos
s'. Explicitly set eval_metric if you'd like to restore the old behavior.
```

```
C:\Users\pandarinath\anaconda3\lib\site-packages\sklearn\utils\validation.py:72: DataConv
ersionWarning: A column-vector y was passed when a 1d array was expected. Please change t
he shape of y to (n_samples, ), for example using ravel().
  return f(**kwargs)
```

Out[43]:

```
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=0.7, gamma=0.1, gpu_id=-1,
              importance_type='gain', interaction_constraints='',
              learning_rate=0.05, max_delta_step=0, max_depth=5,
              min_child_weight=7, missing=None, monotone_constraints='()',
              n_estimators=100, n_jobs=1, nthread=1, num_parallel_tree=1,
              random_state=0, reg_alpha=0, reg_lambda=1, scale_pos_weight=1,
              seed=0, silent=True, subsample=1, tree_method='exact',
              validate_parameters=1, verbosity=None)
```

In [44]:

```
y_pred=classifier.predict(X_test)
```

In [48]:

```
from sklearn.metrics import confusion_matrix,accuracy_score
cm=confusion_matrix(y_test,y_pred)
score=accuracy_score(y_test,y_pred)
```

In [49]:

```
from sklearn.model_selection import cross_val_score
score=cross_val_score(classifier,X_train,y_train.ravel(),cv=10)
```

```
C:\Users\pandarinath\anaconda3\lib\site-packages\xgboost\sklearn.py:888: UserWarning: The
use of label encoder in XGBClassifier is deprecated and will be removed in a future relea
se. To remove this warning, do the following: 1) Pass option use_label_encoder=False when
constructing XGBClassifier object; and 2) Encode your labels (y) as integers starting wit
h 0, i.e. 0, 1, 2, ..., [num_class - 1].
  warnings.warn(label_encoder_deprecation_msg, UserWarning)
```

```
[09:54:50] WARNING: ..\src\learner.cc:541:
Parameters: { silent } might not be used.

  This may not be accurate due to some parameters are only used in language bindings but
  passed down to XGBoost core.  Or some parameters are not used but slip through this
  verification. Please open an issue if you find above cases.


[09:54:50] WARNING: ..\src\learner.cc:1061: Starting in XGBoost 1.3.0, the default evalua
tion metric used with the objective 'binary:logistic' was changed from 'error' to 'loglos
s'. Explicitly set eval_metric if you'd like to restore the old behavior.
[09:54:50] WARNING: ..\src\learner.cc:541:
Parameters: { silent } might not be used.

  This may not be accurate due to some parameters are only used in language bindings but
  passed down to XGBoost core.  Or some parameters are not used but slip through this
  verification. Please open an issue if you find above cases.


[09:54:50] WARNING: ..\src\learner.cc:1061: Starting in XGBoost 1.3.0, the default evalua
tion metric used with the objective 'binary:logistic' was changed from 'error' to 'loglos
s'. Explicitly set eval_metric if you'd like to restore the old behavior.
```

```
[09:54:50] WARNING: ..\src\learner.cc:541:
Parameters: { silent } might not be used.

  This may not be accurate due to some parameters are only used in language bindings but
  passed down to XGBoost core.  Or some parameters are not used but slip through this
  verification. Please open an issue if you find above cases.


[09:54:50] WARNING: ..\src\learner.cc:1061: Starting in XGBoost 1.3.0, the default evalua
tion metric used with the objective 'binary:logistic' was changed from 'error' to 'loglos
s'. Explicitly set eval_metric if you'd like to restore the old behavior.
[09:54:50] WARNING: ..\src\learner.cc:541:
Parameters: { silent } might not be used.

  This may not be accurate due to some parameters are only used in language bindings but
  passed down to XGBoost core.  Or some parameters are not used but slip through this
  verification. Please open an issue if you find above cases.


[09:54:50] WARNING: ..\src\learner.cc:1061: Starting in XGBoost 1.3.0, the default evalua
tion metric used with the objective 'binary:logistic' was changed from 'error' to 'loglos
s'. Explicitly set eval_metric if you'd like to restore the old behavior.
[09:54:50] WARNING: ..\src\learner.cc:541:
Parameters: { silent } might not be used.

  This may not be accurate due to some parameters are only used in language bindings but
  passed down to XGBoost core.  Or some parameters are not used but slip through this
  verification. Please open an issue if you find above cases.


[09:54:50] WARNING: ..\src\learner.cc:1061: Starting in XGBoost 1.3.0, the default evalua
tion metric used with the objective 'binary:logistic' was changed from 'error' to 'loglos
s'. Explicitly set eval_metric if you'd like to restore the old behavior.
[09:54:50] WARNING: ..\src\learner.cc:541:
Parameters: { silent } might not be used.

  This may not be accurate due to some parameters are only used in language bindings but
  passed down to XGBoost core.  Or some parameters are not used but slip through this
  verification. Please open an issue if you find above cases.


[09:54:50] WARNING: ..\src\learner.cc:1061: Starting in XGBoost 1.3.0, the default evalua
tion metric used with the objective 'binary:logistic' was changed from 'error' to 'loglos
s'. Explicitly set eval_metric if you'd like to restore the old behavior.
[09:54:50] WARNING: ..\src\learner.cc:541:
Parameters: { silent } might not be used.

  This may not be accurate due to some parameters are only used in language bindings but
  passed down to XGBoost core.  Or some parameters are not used but slip through this
  verification. Please open an issue if you find above cases.


[09:54:50] WARNING: ..\src\learner.cc:1061: Starting in XGBoost 1.3.0, the default evalua
tion metric used with the objective 'binary:logistic' was changed from 'error' to 'loglos
s'. Explicitly set eval_metric if you'd like to restore the old behavior.
[09:54:50] WARNING: ..\src\learner.cc:541:
Parameters: { silent } might not be used.

  This may not be accurate due to some parameters are only used in language bindings but
  passed down to XGBoost core.  Or some parameters are not used but slip through this
  verification. Please open an issue if you find above cases.


[09:54:50] WARNING: ..\src\learner.cc:1061: Starting in XGBoost 1.3.0, the default evalua
tion metric used with the objective 'binary:logistic' was changed from 'error' to 'loglos
s'. Explicitly set eval_metric if you'd like to restore the old behavior.
[09:54:51] WARNING: ..\src\learner.cc:541:
Parameters: { silent } might not be used.

  This may not be accurate due to some parameters are only used in language bindings but
  passed down to XGBoost core.  Or some parameters are not used but slip through this
  verification. Please open an issue if you find above cases.
```

```
[09:54:51] WARNING: ..\src\learner.cc:1061: Starting in XGBoost 1.3.0, the default evalua
tion metric used with the objective 'binary:logistic' was changed from 'error' to 'loglos
s'. Explicitly set eval_metric if you'd like to restore the old behavior.
[09:54:51] WARNING: ..\src\learner.cc:541:
Parameters: { silent } might not be used.

  This may not be accurate due to some parameters are only used in language bindings but
  passed down to XGBoost core.  Or some parameters are not used but slip through this
  verification. Please open an issue if you find above cases.


[09:54:51] WARNING: ..\src\learner.cc:1061: Starting in XGBoost 1.3.0, the default evalua
tion metric used with the objective 'binary:logistic' was changed from 'error' to 'loglos
s'. Explicitly set eval_metric if you'd like to restore the old behavior.
```

In [50]:

```
score
```

Out[50]:

```
array([0.74074074, 0.81481481, 0.77777778, 0.7962963 , 0.72222222,
       0.74074074, 0.88888889, 0.67924528, 0.83018868, 0.81132075])
```

In [51]:

```
score.mean()
```

Out[51]:

```
0.7802236198462612
```

In [52]:

```
y_pred= classifier.predict(X_test)
```

In [ ]:

```

```