In [3]:
```
import numpy as np
import pandas as pd
import xgboost as xgb
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score
```

In [2]:
```
data = pd.read_csv('pima-indians-diabetes.csv')
df = pd.DataFrame(data=data.values, columns=['Pregnancies', 'Glucose', 'Blood_Pr
df.head()
```

Out[2]:

	Pregnancies	Glucose	Blood_Pressure	Skin_Thickness	Insulin	Bmi	Diabetes_Pedigre
0	1.0	85.0	66.0	29.0	0.0	26.6	0.35
1	8.0	183.0	64.0	0.0	0.0	23.3	0.67
2	1.0	89.0	66.0	23.0	94.0	28.1	0.16
3	0.0	137.0	40.0	35.0	168.0	43.1	2.28
4	5.0	116.0	74.0	0.0	0.0	25.6	0.20

In [4]:
```
def clean_outliers(col) :

    Q1 = df[col].quantile(0.25)
    Q3 = df[col].quantile(0.75)

    IqR = Q3 - Q1

    lower_bound = Q1 - 1.5 * IqR
    upper_bound = Q3 + 1.5 * IqR

    return df[col].clip(lower=lower_bound, upper=upper_bound)

x = df.columns

for i in x :
    df[i] = clean_outliers(i)
```

In [5]:
```
y = df['Prediction'].values.reshape(-1, 1)
```

In [6]:
```
X = df.drop(columns='Prediction')

X = np.hstack((np.ones((X.shape[0],1)),X))
sc = StandardScaler()
X = sc.fit_transform(X)
```

In [7]:
```
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_
```

In [37]:
```
model = xgb.XGBClassifier( objective="binary:logistic",
    n_estimators=1500,
    learning_rate=0.01,
    max_depth=10,
```

```
    subsample=0.9,
    colsample_bytree=0.9,
    eval_metric="logloss"
)

model.fit(x_train, y_train)
y_pred = model.predict(x_test)
```

In [42]:
```
print(f"Accuracy :{accuracy_score(y_test, y_pred) * 100:.2f}%")
```

```
Accuracy :80.52%
```

In [43]:
```
import pickle

with open('XGboost_cla_indians_diabet.pkl', 'wb') as f:
    pickle.dump(model, f)

print("Objet enregistré avec succès !")
```

```
Objet enregistré avec succès !
```