```
!pip install pytorch-tabnet
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

from pytorch_tabnet.tab_model import TabNetClassifier
import torch
from google.colab import files
uploaded = files.upload()
df = pd.read_csv('Raw dataset of heart disease.csv')
X = df.drop('target', axis=1)
y = df['target']

df = pd.read_csv('Raw dataset of heart disease.csv')
df.head()

print(df.info())
print(df.describe())

print(df.isnull().sum())

import seaborn as sns
import matplotlib.pyplot as plt

sns.countplot(x='target', data=df)
plt.show()
```

```
X = df.drop('target', axis=1)
y = df['target']

X = pd.get_dummies(X, drop_first=True)

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
lr_model = LogisticRegression()
lr_model.fit(X_train, y_train)

y_pred_lr = lr_model.predict(X_test)

print("Logistic Regression Accuracy:", accuracy_score(y_test, y_pred_lr))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred_lr))
print("Classification Report:\n", classification_report(y_test, y_pred_lr))

X_train_np = X_train.astype(np.float32)
X_test_np = X_test.astype(np.float32)
y_train_np = y_train.values
y_test_np = y_test.values

tabnet_model = TabNetClassifier()
tabnet_model.fit(
    X_train_np, y_train_np,
    eval_set=[(X_test_np, y_test_np)],
    max_epochs=100,
    patience=10,
```

```
    batch_size=64,
    virtual_batch_size=32,
    num_workers=0,
    drop_last=False
)

y_pred_tabnet = tabnet_model.predict(X_test_np)

print("TabNet Accuracy:", accuracy_score(y_test_np, y_pred_tabnet))
print("Confusion Matrix:\n", confusion_matrix(y_test_np, y_pred_tabnet))
print("Classification Report:\n", classification_report(y_test_np, y_pred_tabnet))

print("Logistic Regression Accuracy:", accuracy_score(y_test, y_pred_lr))
print("TabNet Accuracy:", accuracy_score(y_test_np, y_pred_tabnet))
```