```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import sklearn.datasets
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

df=pd.read_csv("/content/heart.csv")

df.head(5)
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 1025,\n  \"fields\":
[\n    {\n      \"column\": \"age\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 9,\n        \"min\": 29,\n
\"max\": 77,\n        \"num_unique_values\": 41,\n        \"samples\":
[\n          65,\n          50,\n          54\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"sex\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          0,\n          1\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"cp\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 1,\n        \"min\": 0,\n
\"max\": 3,\n        \"num_unique_values\": 4,\n        \"samples\":
[\n          1,\n          3\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"trestbps\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 17,\n        \"min\": 94,\n
\"max\": 200,\n        \"num_unique_values\": 49,\n
\"samples\": [\n          128,\n          172\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"chol\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 51,\n        \"min\": 126,\n
\"max\": 564,\n        \"num_unique_values\": 152,\n
\"samples\": [\n          267,\n          262\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"fbs\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          1,\n          0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"restecg\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 2,\n        \"num_unique_values\": 3,\n        \"samples\":
[\n          1,\n          0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"thalach\",\n      \"properties\": {\n        \"dtype\":
```

```
\"number\",\n        \"std\": 23,\n        \"min\": 71,\n
\"max\": 202,\n        \"num_unique_values\": 91,\n
\"samples\": [\n          180,\n          152\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"exang\",\n      \"properties\": {\
n        \"dtype\": \"number\",\n        \"std\": 0,\n        \"min\":
0,\n        \"max\": 1,\n        \"num_unique_values\": 2,\n
\"samples\": [\n          1,\n          0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"oldpeak\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\":
1.175053255150173,\n        \"min\": 0.0,\n        \"max\": 6.2,\n
\"num_unique_values\": 40,\n        \"samples\": [\n          2.8,\n
0.4\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"slope\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 0,\n        \"min\": 0,\n        \"max\": 2,\n
\"num_unique_values\": 3,\n        \"samples\": [\n          2,\n
0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"ca\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 1,\n        \"min\": 0,\n        \"max\": 4,\n
\"num_unique_values\": 5,\n        \"samples\": [\n          0,\n
4\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"thal\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 0,\n        \"min\": 0,\n        \"max\": 3,\n
\"num_unique_values\": 4,\n        \"samples\": [\n          2,\n
0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"target\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 0,\n        \"min\": 0,\n        \"max\": 1,\n
\"num_unique_values\": 2,\n        \"samples\": [\n          1,\n
0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"df"}
```


```
df.tail(5)
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 5,\n  \"fields\": [\n
{\n      \"column\": \"age\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 5,\n        \"min\": 47,\n
\"max\": 60,\n        \"num_unique_values\": 5,\n        \"samples\":
[\n          60,\n          54,\n          47\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"sex\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          0,\n          1\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
```

```
\"column\": \"cp\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          0,\n          1\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"trestbps\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 12,\n        \"min\": 110,\n
\"max\": 140,\n        \"num_unique_values\": 4,\n        \"samples\":
[\n          125,\n          120\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"chol\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 34,\n        \"min\": 188,\n
\"max\": 275,\n        \"num_unique_values\": 5,\n        \"samples\":
[\n          258,\n          188\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"fbs\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 0,\n        \"num_unique_values\": 1,\n        \"samples\":
[\n          0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"restecg\",\n      \"properties\": {\n        \"dtype\": \"number\",\
n        \"std\": 0,\n        \"min\": 0,\n        \"max\": 1,\n
\"num_unique_values\": 2,\n        \"samples\": [\n          0\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"thalach\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
23,\n        \"min\": 113,\n        \"max\": 164,\n
\"num_unique_values\": 5,\n        \"samples\": [\n          141\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"exang\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 0,\n
\"min\": 0,\n        \"max\": 1,\n        \"num_unique_values\": 2,\n
\"samples\": [\n          0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"oldpeak\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 1.1610340218959994,\n        \"min\":
0.0,\n        \"max\": 2.8,\n        \"num_unique_values\": 4,\n
\"samples\": [\n          2.8\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"slope\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0,\n        \"min\": 1,\n
\"max\": 2,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          1\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"ca\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 0,\n        \"min\": 0,\n        \"max\": 1,\n
\"num_unique_values\": 2,\n        \"samples\": [\n          1\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"thal\",\n      \"properties\":
```

{\n   \"dtype\": \"number\",\n   \"std\": 0,\n   \"min\": 2,\n   \"max\": 3,\n   \"num_unique_values\": 2,\n   \"samples\": [\n   3\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"target\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 0,\n   \"min\": 0,\n   \"max\": 1,\n   \"num_unique_values\": 2,\n   \"samples\": [\n   0\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe"}

```
df.shape
```

```
(1025, 14)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       1025 non-null   int64  
 1   sex       1025 non-null   int64  
 2   cp        1025 non-null   int64  
 3   trestbps  1025 non-null   int64  
 4   chol      1025 non-null   int64  
 5   fbs       1025 non-null   int64  
 6   restecg   1025 non-null   int64  
 7   thalach   1025 non-null   int64  
 8   exang     1025 non-null   int64  
 9   oldpeak   1025 non-null   float64
 10  slope     1025 non-null   int64  
 11  ca        1025 non-null   int64  
 12  thal      1025 non-null   int64  
 13  target    1025 non-null   int64  
dtypes: float64(1), int64(13)
memory usage: 112.2 KB
```

```
df.isnull()
```

{"summary":"{\n \"name\": \"df\",\n \"rows\": 1025,\n \"fields\": [\n   {\n   \"column\": \"age\",\n   \"properties\": {\n   \"dtype\": \"boolean\",\n   \"num_unique_values\": 1,\n   \"samples\": [\n   false\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"sex\",\n   \"properties\": {\n   \"dtype\": \"boolean\",\n   \"num_unique_values\": 1,\n   \"samples\": [\n   false\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"cp\",\n   \"properties\": {\n   \"dtype\": \"boolean\",\n   \"num_unique_values\": 1,\n

\"samples\": [\n   false\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"trestbps\",\n   \"properties\":
{\n   \"dtype\": \"boolean\",\n   \"num_unique_values\": 1,\
n   \"samples\": [\n   false\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"chol\",\n   \"properties\": {\n
\"dtype\": \"boolean\",\n   \"num_unique_values\": 1,\n
\"samples\": [\n   false\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"fbs\",\n   \"properties\": {\n
\"dtype\": \"boolean\",\n   \"num_unique_values\": 1,\n
\"samples\": [\n   false\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"restecg\",\n   \"properties\":
{\n   \"dtype\": \"boolean\",\n   \"num_unique_values\": 1,\
n   \"samples\": [\n   false\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"thalach\",\n   \"properties\":
{\n   \"dtype\": \"boolean\",\n   \"num_unique_values\": 1,\
n   \"samples\": [\n   false\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"exang\",\n   \"properties\": {\
n   \"dtype\": \"boolean\",\n   \"num_unique_values\": 1,\n
\"samples\": [\n   false\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"oldpeak\",\n   \"properties\":
{\n   \"dtype\": \"boolean\",\n   \"num_unique_values\": 1,\
n   \"samples\": [\n   false\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"slope\",\n   \"properties\": {\
n   \"dtype\": \"boolean\",\n   \"num_unique_values\": 1,\n
\"samples\": [\n   false\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"ca\",\n   \"properties\": {\n
\"dtype\": \"boolean\",\n   \"num_unique_values\": 1,\n
\"samples\": [\n   false\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"thal\",\n   \"properties\": {\n
\"dtype\": \"boolean\",\n   \"num_unique_values\": 1,\n
\"samples\": [\n   false\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"target\",\n   \"properties\":
{\n   \"dtype\": \"boolean\",\n   \"num_unique_values\": 1,\
n   \"samples\": [\n   false\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   }\n]\n}","type":"dataframe"}

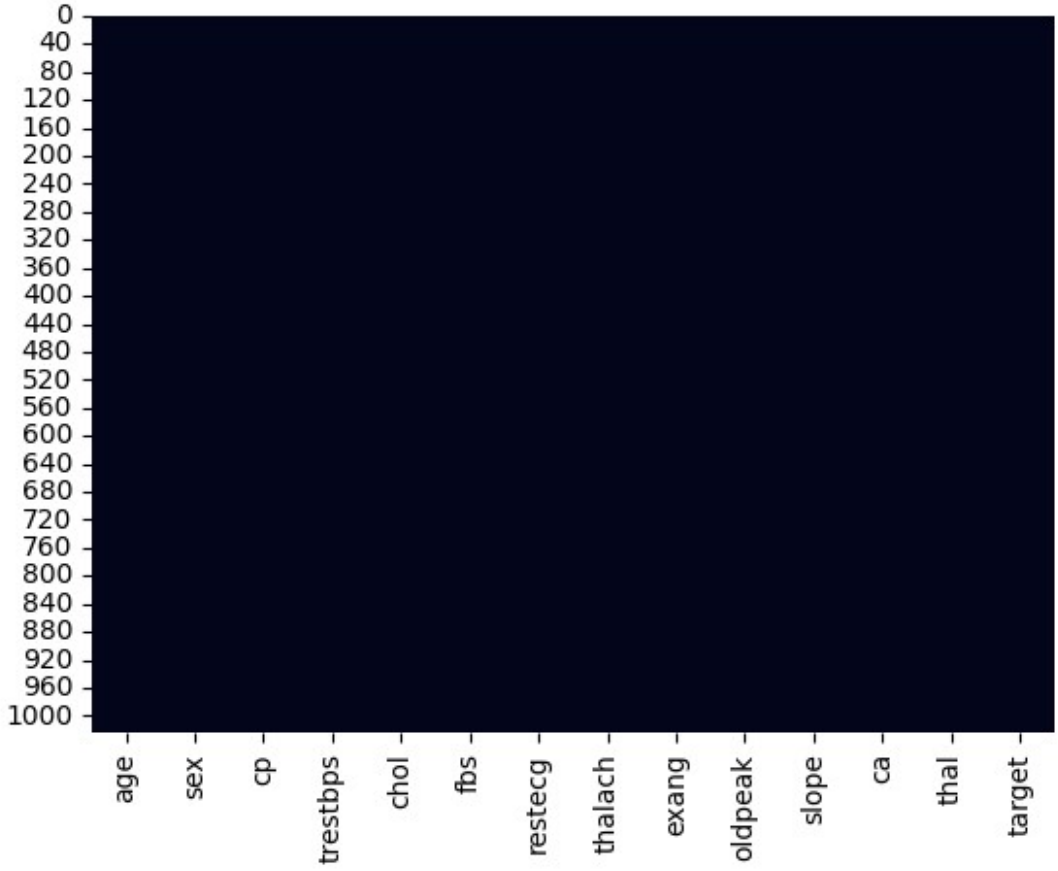
```
df.isnull().sum()
```

```
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
target      0
dtype: int64
```

```
df.isnull().sum().sum()
```

```
np.int64(0)
```

```
sns.heatmap(df.isnull(),cbar=False)
plt.show()
```



```
df.describe()
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 8,\n  \"fields\": [\n
{\n      \"column\": \"age\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 346.1150023272597,\n
\"min\": 9.072290233244281,\n        \"max\": 1025.0,\n
\"num_unique_values\": 8,\n        \"samples\": [\n
54.43414634146342,\n          56.0,\n          1025.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"sex\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 362.1825587675373,\n
\"min\": 0.0,\n        \"max\": 1025.0,\n
\"num_unique_values\": 5,\n        \"samples\": [\n
0.6956097560975609,\n          1.0,\n          0.4603733241196503\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"cp\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 361.9909299680307,\n
\"min\": 0.0,\n        \"max\": 1025.0,\n
\"num_unique_values\": 7,\n        \"samples\": [\n          1025.0,\n
0.9424390243902439,\n          2.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"trestbps\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\":
324.3475475669103,\n        \"min\": 17.516718005376408,\n
\"max\": 1025.0,\n        \"num_unique_values\": 8,\n
\"samples\": [\n          131.61170731707318,\n          130.0,\n
1025.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"chol\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 313.5134241805058,\n        \"min\": 51.59251020618206,\n
\"max\": 1025.0,\n        \"num_unique_values\": 8,\n
\"samples\": [\n          246.0,\n          240.0,\n          1025.0\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"fbs\",\n      \"properties\": {\
n        \"dtype\": \"number\",\n        \"std\": 362.3163339641884,\n
\"min\": 0.0,\n        \"max\": 1025.0,\n
\"num_unique_values\": 5,\n        \"samples\": [\n
0.14926829268292682,\n          1.0,\n          0.3565266897271594\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"restecg\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
362.1373492815066,\n        \"min\": 0.0,\n        \"max\": 1025.0,\n
\"num_unique_values\": 6,\n        \"samples\": [\n          1025.0,\n
0.5297560975609756,\n          2.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"thalach\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\":
322.1492006877167,\n        \"min\": 23.005723745977196,\n
\"max\": 1025.0,\n        \"num_unique_values\": 8,\n
\"samples\": [\n          149.11414634146342,\n          152.0,\n
```

1025.0\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"exang\",\n   \"properties\": {\n   \"dtype\": \"number\",\n
\"std\": 362.2505698153287,\n   \"min\": 0.0,\n   \"max\":
1025.0,\n   \"num_unique_values\": 5,\n   \"samples\": [\n
0.33658536585365856,\n   1.0,\n   0.4727723760037095\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   },\n   {\n   \"column\": \"oldpeak\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
361.83970171608024,\n   \"min\": 0.0,\n   \"max\": 1025.0,\n
\"num_unique_values\": 7,\n   \"samples\": [\n   1025.0,\n
1.0715121951219515,\n   1.8\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"slope\",\n   \"properties\": {\
n   \"dtype\": \"number\",\n   \"std\": 361.9886284380937,\n
\"min\": 0.0,\n   \"max\": 1025.0,\n
\"num_unique_values\": 6,\n   \"samples\": [\n   1025.0,\n
1.3853658536585365,\n   2.0\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"ca\",\n   \"properties\": {\n
\"dtype\": \"number\",\n   \"std\": 362.05191221025626,\n
\"min\": 0.0,\n   \"max\": 1025.0,\n
\"num_unique_values\": 6,\n   \"samples\": [\n   1025.0,\n
0.7541463414634146,\n   4.0\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"thal\",\n   \"properties\": {\n
\"dtype\": \"number\",\n   \"std\": 361.7399759382844,\n
\"min\": 0.0,\n   \"max\": 1025.0,\n
\"num_unique_values\": 6,\n   \"samples\": [\n   1025.0,\n
2.323902439024390243902439,\n   3.0\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"target\",\n   \"properties\":
{\n   \"dtype\": \"number\",\n   \"std\":
362.1897651954653,\n   \"min\": 0.0,\n   \"max\": 1025.0,\n
\"num_unique_values\": 5,\n   \"samples\": [\n
0.5131707317073171,\n   1.0,\n   0.5000704980788051\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   }\n]\n}","type":"dataframe"}

```
# Checking the distribution of target  variables
df['target'].value_counts()
```

```
target
1    526
0    499
Name: count, dtype: int64
```

0 ------> Defective From Heart ❤ Disease

1 ------> Healthy ♥

Spliting The Feature And Target Columns From Dataset's

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

print(X)
```

```
      age  sex  cp  trestbps  chol  ...  exang  oldpeak  slope  ca
thal
0      52    1   0       125   212  ...      0      1.0      2   2
3
1      53    1   0       140   203  ...      1      3.1      0   0
3
2      70    1   0       145   174  ...      1      2.6      0   0
3
3      61    1   0       148   203  ...      0      0.0      2   1
3
4      62    0   0       138   294  ...      0      1.9      1   3
2
...   ...  ...  ..       ...   ...  ...    ...      ...    ...  ..   .
..
1020   59    1   1       140   221  ...      1      0.0      2   0
2
1021   60    1   0       125   258  ...      1      2.8      1   1
3
1022   47    1   0       110   275  ...      1      1.0      1   1
2
1023   50    0   0       110   254  ...      0      0.0      2   0
2
1024   54    1   0       120   188  ...      0      1.4      1   1
3

[1025 rows x 13 columns]
```

```
print(Y)
```

```
0       0
1       0
2       0
3       0
```

```
4       0
       ..
1020    1
1021    0
1022    0
1023    1
1024    0
Name: target, Length: 1025, dtype: int64
```

Spliting The Data Training Data And Testing Data

```
# Train Test Split
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2,strat
ify=Y,random_state=2)

print(X.shape,X_train.shape,X_test.shape)
```

```
(1025, 13) (820, 13) (205, 13)
```

```
X_train
```


```
{"summary":"{\n  \"name\": \"X_train\",\n  \"rows\": 820,\n
\"fields\": [\n    {\n      \"column\": \"age\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
9,\n        \"min\": 29,\n        \"max\": 77,\n
\"num_unique_values\": 40,\n        \"samples\": [\n          39,\n
54,\n          60\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"sex\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 0,\n        \"min\": 0,\n        \"max\": 1,\n
\"num_unique_values\": 2,\n        \"samples\": [\n          0,\n
1\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"cp\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 1,\n        \"min\": 0,\n        \"max\": 3,\n
\"num_unique_values\": 4,\n        \"samples\": [\n          2,\n
1\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"trestbps\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 17,\n        \"min\": 94,\n
\"max\": 200,\n        \"num_unique_values\": 49,\n
\"samples\": [\n          108,\n          155\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"chol\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 51,\n        \"min\": 126,\n
\"max\": 564,\n        \"num_unique_values\": 152,\n
```

\"samples\": [\n   205,\n   141\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"fbs\",\n   \"properties\": {\n
\"dtype\": \"number\",\n   \"std\": 0,\n   \"min\": 0,\n
\"max\": 1,\n   \"num_unique_values\": 2,\n   \"samples\":
[\n   0,\n   1\n   ],\n   \"semantic_type\":
\"\",\n   \"description\": \"\"\n   }\n   },\n   {\n
\"column\": \"restecg\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 0,\n   \"min\": 0,\n
\"max\": 2,\n   \"num_unique_values\": 3,\n   \"samples\":
[\n   1,\n   0\n   ],\n   \"semantic_type\":
\"\",\n   \"description\": \"\"\n   }\n   },\n   {\n
\"column\": \"thalach\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 23,\n   \"min\": 71,\n
\"max\": 202,\n   \"num_unique_values\": 90,\n
\"samples\": [\n   106,\n   165\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"exang\",\n   \"properties\": {\
n   \"dtype\": \"number\",\n   \"std\": 0,\n   \"min\":
0,\n   \"max\": 1,\n   \"num_unique_values\": 2,\n
\"samples\": [\n   0,\n   1\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"oldpeak\",\n   \"properties\":
{\n   \"dtype\": \"number\",\n   \"std\":
1.1933667373729135,\n   \"min\": 0.0,\n   \"max\": 6.2,\n
\"num_unique_values\": 40,\n   \"samples\": [\n   2.2,\n
4.2\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"slope\",\n   \"properties\": {\n   \"dtype\": \"number\",\n
\"std\": 0,\n   \"min\": 0,\n   \"max\": 2,\n
\"num_unique_values\": 3,\n   \"samples\": [\n   1,\n
2\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"ca\",\n   \"properties\": {\n   \"dtype\": \"number\",\n
\"std\": 1,\n   \"min\": 0,\n   \"max\": 4,\n
\"num_unique_values\": 5,\n   \"samples\": [\n   1,\n
4\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"thal\",\n   \"properties\": {\n   \"dtype\": \"number\",\n
\"std\": 0,\n   \"min\": 0,\n   \"max\": 3,\n
\"num_unique_values\": 4,\n   \"samples\": [\n   3,\n
1\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   }\n]\
n}","type":"dataframe","variable_name":"X_train"}

Y_train

14	0
586	0
16	1

```
462     1
186     0
       ..
221     0
906     1
1002    0
966     0
191     1
Name: target, Length: 820, dtype: int64
```

Model Training

```
Heart_Disease_Model=LogisticRegression()

# Training The Logistics Model With Traning Model
Heart_Disease_Model.fit(X_train,Y_train)
```

```
/usr/local/lib/python3.11/dist-packages/sklearn/linear_model/
_logistic.py:465: ConvergenceWarning: lbfgs failed to converge
(status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as
shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(

LogisticRegression()
```

Model Evaluation

```
# Accuracy Score On Training Data
X_train_prediction=Heart_Disease_Model.predict(X_train)
training_data_accuracy=accuracy_score(X_train_prediction,Y_train)

print('Accuracy On Training Data : ',training_data_accuracy)
```

```
Accuracy On Training Data :  0.8524390243902439
```

```
# Accuracy Score On Test Data
X_test_prediction=Heart_Disease_Model.predict(X_test)
test_data_accuracy=accuracy_score(X_test_prediction,Y_test)

print('Accuracy On Test Data : ',test_data_accuracy)
```

```
Accuracy On Test Data :  0.8048780487804879
```

Make A Predictive System

```
Input_data=(40,0,1,150,207,0,0,176,0,1.6,2,0,2)

# Change The Input Data To Numpy Array
input_data_as_numpy_array=np.asarray(Input_data)

# Reshape The Numpy Array
input_data_reshaped=input_data_as_numpy_array.reshape(1,-1)

prediction=Heart_Disease_Model.predict(input_data_reshaped)
print(prediction)

if(prediction[0]==0):
  print('The Person Does Not Have A Heart Disease')
else:
  print('The Person Have A Heart Disease')
```

```
[1]
The Person Have A Heart Disease
```

```
import joblib

joblib.dump(Heart_Disease_Model, 'heart_disease_model.pkl')
```

```
['heart_disease_model.pkl']
```