Importing the Dependencies

import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

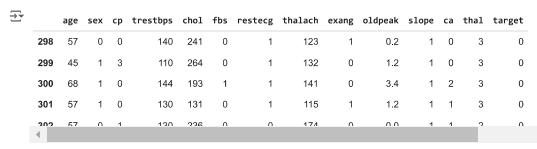
Data Collection and Processing

loading the csv data to a Pandas DataFrame
heart_data = pd.read_csv('/content/heart_disease_data.csv')

print first 5 rows of the dataset
heart_data.head()

→		age	sex	ср	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
	0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
	1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
	2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
	3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
	A	57	0	^	120	25/	Λ	1	163	1	0.6	2	^	?	1

print last 5 rows of the dataset
heart_data.tail()



number of rows and columns in the dataset
heart_data.shape

→ (303, 14)

getting some info about the data
heart_data.info()

<class 'pandas.core.frame.DataFrame'> RangeIndex: 303 entries, 0 to 302 Data columns (total 14 columns): # Column Non-Null Count Dtype 0 age 303 non-null int64 303 non-null int64 1 sex 2 303 non-null int64 3 trestbps 303 non-null int64 4 303 non-null int64 cho1 5 fbs 303 non-null int64 303 non-null int64 restecg 303 non-null int64 thalach 8 303 non-null int64 exang 9 oldpeak 303 non-null float64 10 slope 303 non-null int64 303 non-null int64 11 ca 12 thal 303 non-null int64 target 303 non-null int64 dtypes: float64(1), int64(13) memory usage: 33.3 KB



™CAfee | WebAdvisor

Your download's being scanned. We'll let you know if there's an issue.

+ Code

+ Text

×

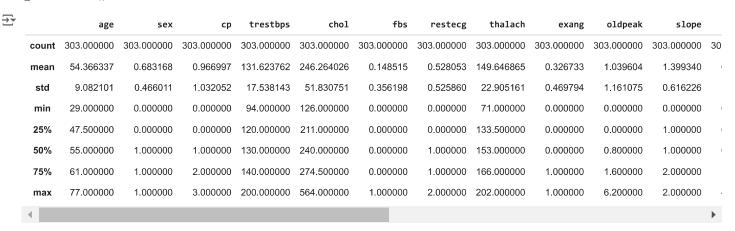
checking for missing values
heart_data.isnull().sum()



target

0

statistical measures about the data
heart_data.describe()



checking the distribution of Target Variable
heart_data['target'].value_counts()



count

 target

 1
 165

 0
 138

Splitting the Features and Target

X = heart_data.drop(columns='target', axis=1)

Y = heart_data['target']

print(X)

		age	sex	ср	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	\
	0	63	1	3	145	233	1	0	150	0	2.3	
	1	37	1	2	130	250	0	1	187	0	3.5	
	2	41	0	1	130	204	0	0	172	0	1.4	



Model Evaluation

LogisticRegression ① ②

LogisticRegression()

Accuracy Score



```
# accuracy on training data
X_train_prediction = 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 : 0.8512396694214877
# accuracy on test data
X_test_prediction = 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 : 0.819672131147541
Building a Predictive System
input_data = (62,0,0,140,268,0,0,160,0,3.6,0,2,2)
# change the input data to a numpy array
input_data_as_numpy_array= np.asarray(input_data)
# reshape the numpy array as we are predicting for only on instance
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)
prediction = model.predict(input_data_reshaped)
print(prediction)
if (prediction[0]== 0):
 print('The Person does not have a Heart Disease')
 print('The Person has Heart Disease')
<del>_</del>_
    [0]
     The Person does not have a Heart Disease
     /usr/local/lib/python3.10/dist-packages/sklearn/base.py:493: UserWarning: X does not have valid feature names, but LogisticRegression wa
       warnings.warn(
```

