```
print("Accuracy (Test) score of SVM ", sv.score(X_test,Y_test)*100)
print("Accuracy score of SVM ", accuracy_score(Y_test,sv_pred)*100)

# Train & Test Scores of Decision Tree
print("Accuracy (Train) score of Decision Tree
",dt.score(X_train,Y_train)*100)
print("Accuracy (Test) score of Decision Tree ", dt.score(X_test,Y_test)*100)
print("Accuracy score of Decision Tree ", accuracy_score(Y_test,dt_pred)*100)

# Train & Test Scores of Random Forest
print("Accuracy (Train) score of Random Forest
",rf.score(X_train,Y_train)*100)
print("Accuracy (Test) score of Random Forest ", rf.score(X_test,Y_test)*100)
print("Accuracy score of Random Forest ", accuracy_score(Y_test,rf_pred)*100)
```

Step 10: Building a Predictive System

```
input_data = (63,1,3,145,233,1,0,150,0,2.3,0,0,1)

# 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 = dt.predict(input_data_reshaped)
print(prediction)

if (prediction[0]== 0):
  print('The Person does not have a Heart Disease')
else:
  print('The Person has Heart Disease')
```

Step 11 : Saving the trained model

```
import pickle
import pickle
filename = 'heart_disease_model.sav'
pickle.dump(lr,open(filename,'wb'))
```