PREDICTION OF HEART DISEASES USING MACHINE LEARNING MODELS

Anjali Raman

Department of Computer Science and Engineering

University of North Texas

Introduction

The human heart, is an essential part of the human body. There are several indications and symptoms of heart disease. These symptoms include chest pain, shortness of breath, chest pressure, angina, arm and limb pain and numbness, abnormal heartbeats, dry or chronic coughing, rashes, or other unusual skin patches, and so on. Cardiovascular disease is the leading cause of mortality. Heart disorders can manifest itself in a variety of ways. They include arrhythmia, coronary artery disease, heart attack, chest discomfort, stroke, irregular heartbeat, and congenital heart disease. The major causes of coronary disease include high blood pressure, high cholesterol, and a quick pulse. Bad lifestyle choices, such as a poor diet, insufficient exercise, a high BMI, and smoking, all considerably raise the risk of heart disease. Furthermore, your family's medical history may have a role in the development of heart disease.

Every year, 18 million people around the globe die as a result of heart disease. According to WHO 17.5 million people died from heart disease globally in 2005, accounting for 31% of all fatalities. Nonetheless, the fatality toll climbs dramatically from year to year. It is expected to top 23.6 million by 2030.

Using machine learning to the problem of forecasting heart illness can provide several major benefits. To begin, technology can assist doctors in making more accurate and timely diagnoses. It can help identify those who are at high risk of developing heart disease, enabling for early intervention and preventative care. Machine learning may save time and money by automating critical components of the diagnostic process, enabling medical practitioners to focus on more challenging cases. Moreover, machine learning may help determine which diagnostic tests are most effective in predicting cardiac illness, allowing for more targeted and efficient resource utilization. Our machine learning technology will ultimately benefit medical experts, healthcare providers, and patients. Forecasting cardiac abnormalities accurately using machine learning can have important consequences for improving patient outcomes, decreasing healthcare costs, and optimizing resource allocation.

Literature Review

When using machine learning to predict cardiac illness, the interpretability of the models is critical to consider. While the research cited suggest that machine learning algorithms have the potential to improve diagnostic accuracy, they do not explain how the models make predictions. Due of this lack of transparency, healthcare practitioners may struggle to trust and employ the models in clinical practice. As a result, my study will centre on developing interpretable models that can provide insights into the basic factors impacting projections. This can boost the models' trustworthiness and assist healthcare practitioners to make more informed decisions when diagnosing and treating heart disease.

Paper-1:

"Comparison of Machine Learning Models for Prediction of Heart Disease" by Sravani Avula. (2019)

This paper investigated the efficacy of several machine learning algorithms for predicting heart disease using the "Predicting Heart Disease Risk Using Clinical Variables" dataset. The approaches used were decision trees, random forests, support vector machines, logistic regression, and k-nearest neighbours. Also, the models' accuracy, sensitivity, specificity, and AUC-ROC were all assessed using 10-fold cross-validation. The random forest technique outperformed the other algorithms in terms of accuracy, sensitivity, specificity, and AUC-ROC. The random forest algorithm, according to the study's findings, is the most effective method for predicting heart disease based on clinical criteria.

Paper-2:

"Predicting heart disease using ensemble of feature selection and machine learning algorithms" by N. Natarajan and M. Priya (2017):

This study described an ensemble method for predicting cardiac disease that incorporates feature selection and machine learning techniques. The Correlation-based Feature Selection (CFS) method was used to choose the most significant features, and a variety of machine learning techniques, including Decision Tree, Naive Bayes, and Support Vector Machine, were employed to predict the existence of heart disease (SVM). According to the study, the ensemble technique outperformed individual machine learning algorithms in predicting heart disease.

Paper-3:

"A hybrid feature selection approach for predicting heart disease using clinical data" by N. Alshamlan and A. Badr (2016):

Using clinical data, this study developed a hybrid feature selection technique for predicting heart disease. The chi-squared statistic and the Relief-F algorithms were used to find the most significant features and reduce the dimensionality of the dataset. The proposed method outperformed existing feature selection approaches such as Principal Component Analysis (PCA) and Correlation-based Feature Selection in predicting heart disease, according to the research (CFS).

Paper-4:

"An ensemble-based machine learning approach for heart disease prediction" by Alomari. (2021)

This work provides an ensemble-based machine learning strategy for forecasting the risk of heart disease using decision trees, support vector machines, and logistic regression. The suggested strategy outperformed other individual techniques with an accuracy of 89.67%. The study also discovered that the most critical factors in determining the risk of heart disease were age, maximum heart rate reached, and exercise-induced angina.

Data Exploration

The dataset used for this project “Predicting heart disease risk using clinical variables” contains various clinical and demographic information about the patients from different parts of the world such as India, United states of America, Germany. The age groups between 18 to 90 years are studied for the dataset. The levels of cholesterol, diabetes, blood pressure levels are taken into account resulting in 9,670 observations.

Generally, the prediction of heart disease can be classified into two classes, they are “Positive” and “Negative” for the existence of this coronary disease. This data would determine whether the classes are training data or testing data. If the data is huge, the dataset can be categorized based on age, sex, BP, cholesterol, ECG etc.

According to the dataset, there are 303 records that can be considered. The features in the dataset are as follows:

1. Age of the patient (Age in years).
2. Sex of the patients (1-male, 0- female).
3. Type of Chest Pain (1- typical angina, 2- atypical angina, 3- non anginal pain, 4- asymptomatic).
4. Blood Pressure
5. Cholesterol Levels (in mg/dl).
6. Fasting blood sugar (>120mg/dl; 1- True, 0-False).
7. EKG results (0- normal, 1- ST-T wave abnormality, 2- probable or definite left ventricular hypertrophy).
8. Max Heart Rate (beats per min).
9. Exercise Angina (1- yes, 0- No).
10. ST Depression
11. Slope of ST Depression (1- unsloping, 2- flat, 3- down sloping).
12. No. of vessels based on fluoroscopy (0-3).
13. Thallium levels (3- normal, 6- fixed defect, 7- reversible defect).
14. Diagnosis of Heart disease (1- yes, 0- no).

The interesting part about dealing with the dataset is that it contains small number of instances when compared with another medical datasets. This dataset is mainly dedicated to clinical research work with variety of clinical terms such as chest pain type, blood pressure levels, cholesterol levels heart rate levels etc, to predict the presence of heart disease. Therefore, this can help the researchers in a better way. Further, this dataset also provides information about the presence of heart disease which in turn helps in evaluating machine learning models for the future predictions.

PROJECT DELIVERABLE-2

Evaluation Metrics:

The evaluation metrics are generally used to determine the efficiency of the model. The evaluation metrics gives us numerical measures determining the performance of the model. Evaluation metrics can be evaluated using various metrics based on the dataset considered. the evaluation metrics also depend on the baseline of the data. The baseline can be of two type. They are:

- Classification
- Regression

Here, in this project classification is considered to be the baseline. Therefore, the evaluation metrics that needs to calculated can be given as

- Accuracy: the accuracy of data determines the predictions of model for the data
- Pression: the precisions give true or false predictions for the given data.
- F1 score: F1 score gives the mean value for the data and determines the efficiency of the model.
- Recall: the recall gives the percentage value of the model

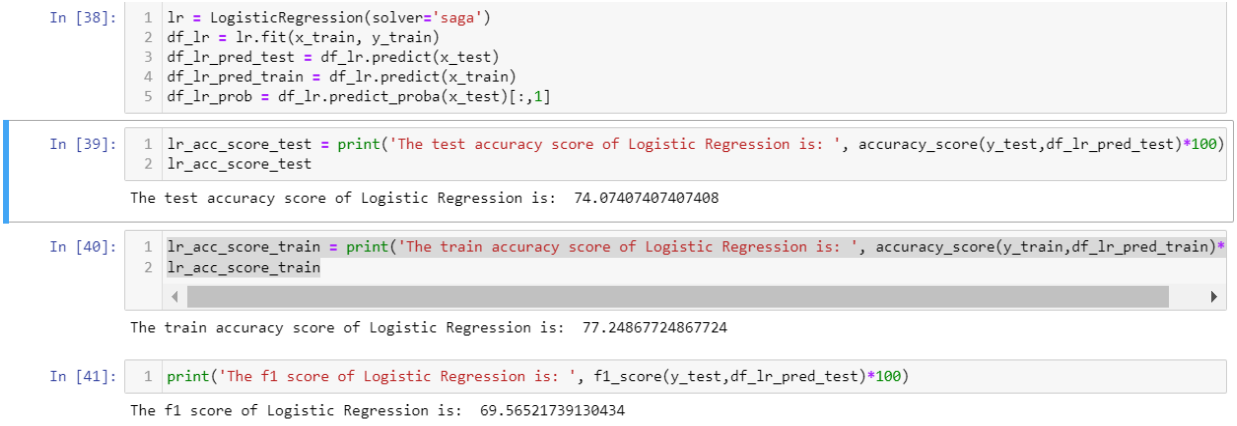
Baseline:

For the dataset, the baseline is considered to be Classifier which focuses on the Presence or Absence of Heart Disease. The observations are based on many factors such as Age, Sex, Cholesterol levels, BP levels, Heart Rate etc. In classification model, multiple evaluation metrics can be evaluated for better understanding. In this case, the objective is to evaluate accuracy and F1 score using different machine learning techniques determines the best fit for prediction of heart disease. Also, the majority class for the dataset needs to be evaluated as we are considering classification as baseline. In this project, I will try to determine the accuracy and F1 score for different machine learning algorithms such as Logistic Regression, Decision Tree, Random Forrest and KNN Algorithm.

Analysis of Classification model:

The classification for the data is evaluated using various machine learning algorithms such as Logistic Regression, Decision Tree, Random Forrest and KNN algorithm. The evaluation outputs are as follows

Logistic Regression:



```
In [38]:
lr = LogisticRegression(solver='saga')
df_lr = lr.fit(x_train, y_train)
df_lr_pred_test = df_lr.predict(x_test)
df_lr_pred_train = df_lr.predict(x_train)
df_lr_prob = df_lr.predict_proba(x_test)[:,1]
```

```
In [39]:
lr_acc_score_test = print('The test accuracy score of Logistic Regression is: ', accuracy_score(y_test,df_lr_pred_test)*100)
lr_acc_score_test
```

```
The test accuracy score of Logistic Regression is:  74.07407407407408
```

```
In [40]:
lr_acc_score_train = print('The train accuracy score of Logistic Regression is: ', accuracy_score(y_train,df_lr_pred_train)*
lr_acc_score_train
```

```
The train accuracy score of Logistic Regression is:  77.24867724867724
```

```
In [41]:
print('The f1 score of Logistic Regression is: ', f1_score(y_test,df_lr_pred_test)*100)
```

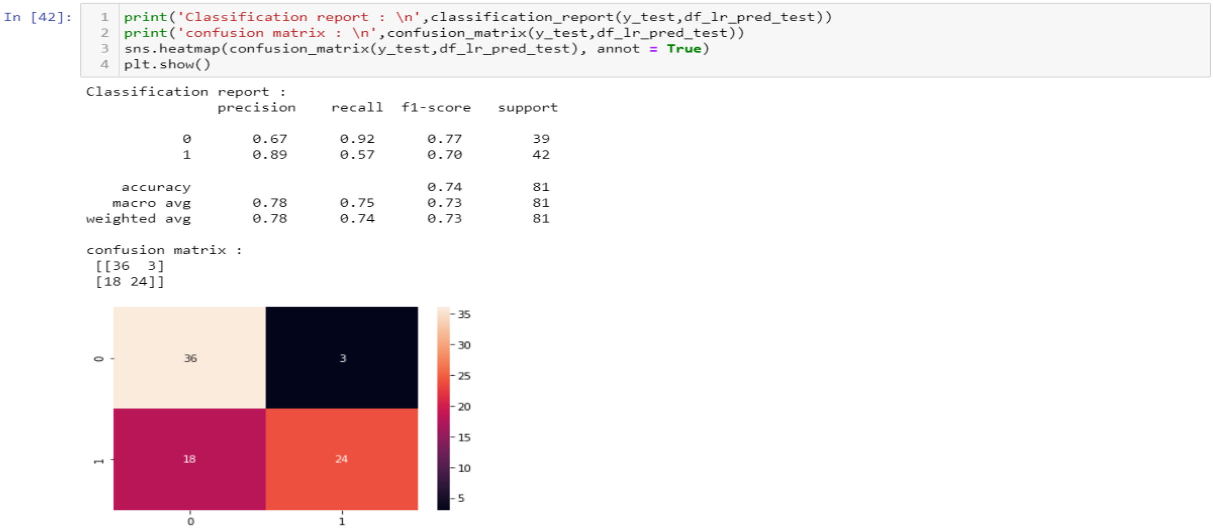
```
The f1 score of Logistic Regression is:  69.56521739130434
```

The above fig. shows the evaluation results of the machine learning algorithm Logistic Regression

The accuracy of training data is 77.2

The accuracy of testing data is 74.07

The F1 score is 69.56

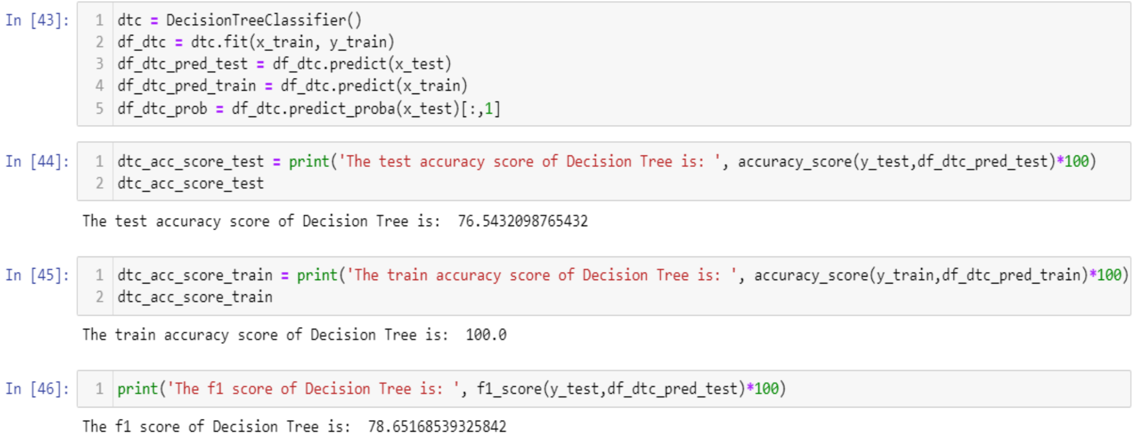


```
In [42]:
print('Classification report : \n',classification_report(y_test,df_lr_pred_test))
print('confusion matrix : \n',confusion_matrix(y_test,df_lr_pred_test))
sns.heatmap(confusion_matrix(y_test,df_lr_pred_test), annot = True)
plt.show()
```

```
Classification report :
               precision    recall  f1-score   support

           0       0.67      0.92      0.77        39
           1       0.89      0.57      0.70        42

    accuracy                           0.74        81
   macro avg       0.78      0.75      0.73        81
weighted avg       0.78      0.74      0.73        81

confusion matrix :
 [[36  3]
 [18 24]]
```

The above fig. shows the analysis of classification report of Logistic Regression using confusion matrix where the **Accuracy is 0.74**

Decision Tree:



In [43]:
```
dtc = DecisionTreeClassifier()
df_dtc = dtc.fit(x_train, y_train)
df_dtc_pred_test = df_dtc.predict(x_test)
df_dtc_pred_train = df_dtc.predict(x_train)
df_dtc_prob = df_dtc.predict_proba(x_test)[:,1]
```

In [44]:
```
dtc_acc_score_test = print('The test accuracy score of Decision Tree is: ', accuracy_score(y_test,df_dtc_pred_test)*100)
dtc_acc_score_test
```
```
The test accuracy score of Decision Tree is:  76.5432098765432
```

In [45]:
```
dtc_acc_score_train = print('The train accuracy score of Decision Tree is: ', accuracy_score(y_train,df_dtc_pred_train)*100)
dtc_acc_score_train
```
```
The train accuracy score of Decision Tree is:  100.0
```

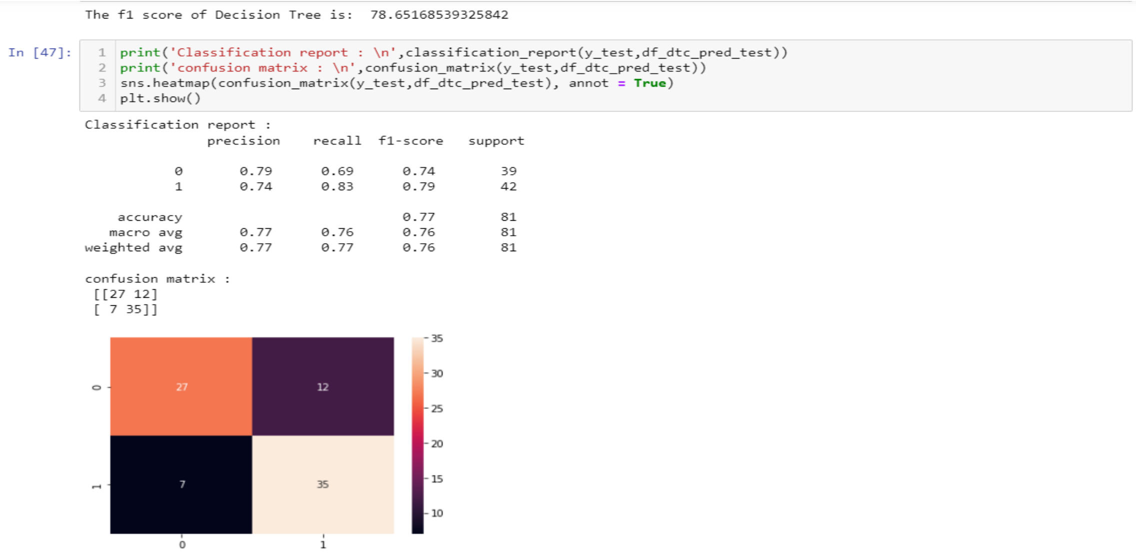
In [46]:
```
print('The f1 score of Decision Tree is: ', f1_score(y_test,df_dtc_pred_test)*100)
```
```
The f1 score of Decision Tree is:  78.65168539325842
```

The above fig. shows the evaluation results of the machine learning algorithm Decision Tree

The accuracy of training data is 100

The accuracy of testing data is 76.5

The F1 score is 78.65

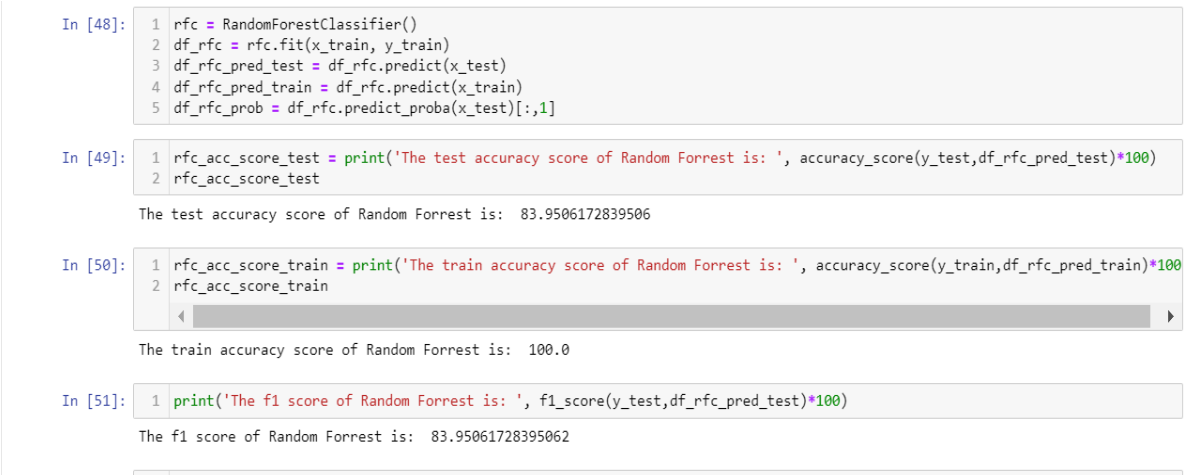


```
The f1 score of Decision Tree is:  78.65168539325842
```

In [47]:
```
print('Classification report : \n',classification_report(y_test,df_dtc_pred_test))
print('confusion matrix : \n',confusion_matrix(y_test,df_dtc_pred_test))
sns.heatmap(confusion_matrix(y_test,df_dtc_pred_test), annot = True)
plt.show()
```
```
Classification report :
               precision    recall  f1-score   support

           0       0.79      0.69      0.74        39
           1       0.74      0.83      0.79        42

    accuracy                           0.77        81
   macro avg       0.77      0.76      0.76        81
weighted avg       0.77      0.77      0.76        81

confusion matrix :
 [[27 12]
 [ 7 35]]
```

The above fig. shows the analysis of classification report for Decision Tree algorithm using confusion matrix where the **Accuracy is 0.77**

Random Forrest:



```
In [48]:
rfc = RandomForestClassifier()
df_rfc = rfc.fit(x_train, y_train)
df_rfc_pred_test = df_rfc.predict(x_test)
df_rfc_pred_train = df_rfc.predict(x_train)
df_rfc_prob = df_rfc.predict_proba(x_test)[:,1]
```

```
In [49]:
rfc_acc_score_test = print('The test accuracy score of Random Forrest is: ', accuracy_score(y_test,df_rfc_pred_test)*100)
rfc_acc_score_test
```

```
The test accuracy score of Random Forrest is:  83.9506172839506
```

```
In [50]:
rfc_acc_score_train = print('The train accuracy score of Random Forrest is: ', accuracy_score(y_train,df_rfc_pred_train)*100
rfc_acc_score_train
```

```
The train accuracy score of Random Forrest is:  100.0
```

```
In [51]:
print('The f1 score of Random Forrest is: ', f1_score(y_test,df_rfc_pred_test)*100)
```

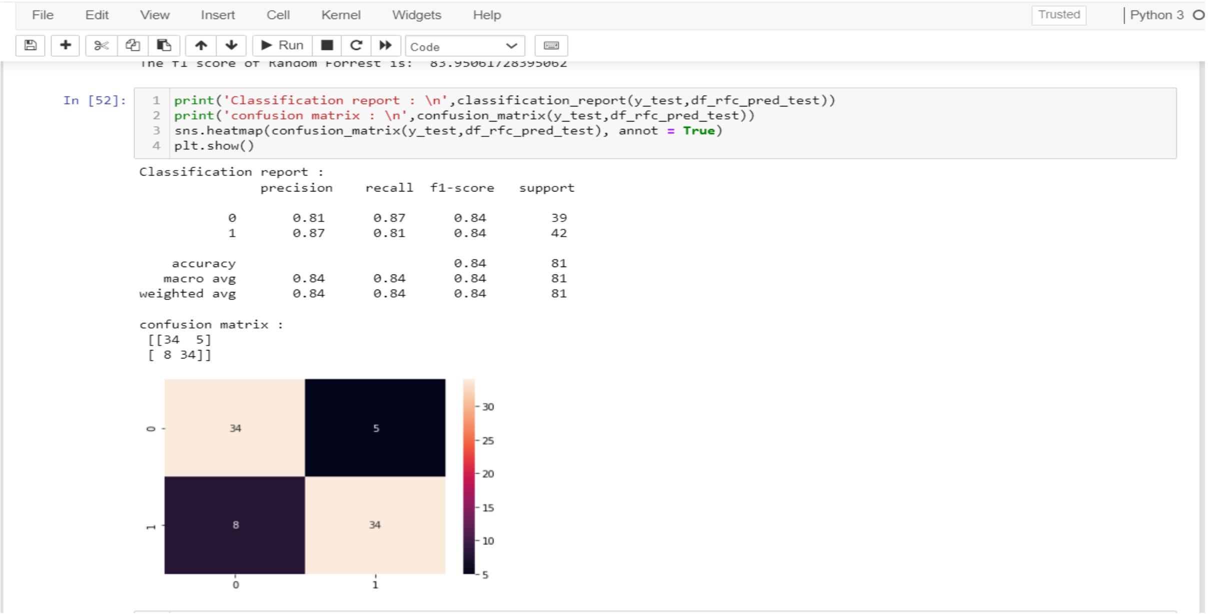
```
The f1 score of Random Forrest is:  83.95061728395062
```

The above fig. shows the evaluation results of the machine learning algorithm Decision Tree

The accuracy of training data is 100

The accuracy of testing data is 83.95

The F1 score is 83.95



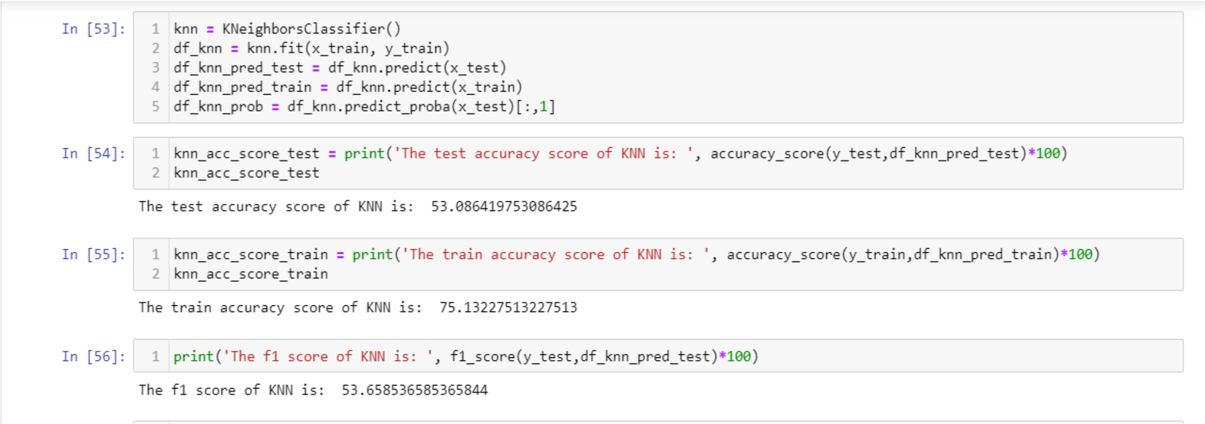
```
The f1 score of Random Forrest is:  83.95061728395062
```

```
In [52]:
print('Classification report : \n',classification_report(y_test,df_rfc_pred_test))
print('confusion matrix : \n',confusion_matrix(y_test,df_rfc_pred_test))
sns.heatmap(confusion_matrix(y_test,df_rfc_pred_test), annot = True)
plt.show()
```

```
Classification report :
               precision    recall  f1-score   support

           0       0.81      0.87      0.84        39
           1       0.87      0.81      0.84        42

    accuracy                           0.84        81
   macro avg       0.84      0.84      0.84        81
weighted avg       0.84      0.84      0.84        81

confusion matrix :
 [[34  5]
 [ 8 34]]
```

The above fig. shows the analysis of classification report of Random Forrest using confusion matrix where the **Accuracy is 0.84**

KNN Algorithm:



```
In [53]:
knn = KNeighborsClassifier()
df_knn = knn.fit(x_train, y_train)
df_knn_pred_test = df_knn.predict(x_test)
df_knn_pred_train = df_knn.predict(x_train)
df_knn_prob = df_knn.predict_proba(x_test)[:,1]
```

```
In [54]:
knn_acc_score_test = print('The test accuracy score of KNN is: ', accuracy_score(y_test,df_knn_pred_test)*100)
knn_acc_score_test
```

```
The test accuracy score of KNN is:  53.086419753086425
```

```
In [55]:
knn_acc_score_train = print('The train accuracy score of KNN is: ', accuracy_score(y_train,df_knn_pred_train)*100)
knn_acc_score_train
```

```
The train accuracy score of KNN is:  75.13227513227513
```

```
In [56]:
print('The f1 score of KNN is: ', f1_score(y_test,df_knn_pred_test)*100)
```

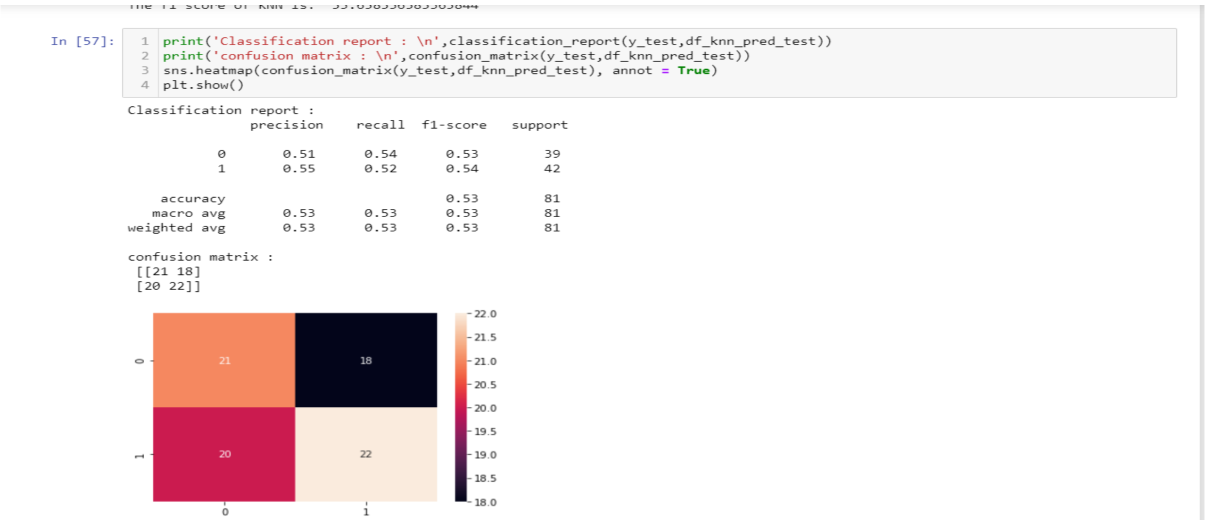
```
The f1 score of KNN is:  53.658536585365844
```

The above fig. shows the evaluation results of the machine learning algorithm Decision Tree

The accuracy of training data is 75.13

The accuracy of testing data is 53.08

The F1 score is 53.65



```
The f1 score of KNN is:  53.658536585365844
```

```
In [57]:
print('Classification report : \n',classification_report(y_test,df_knn_pred_test))
print('confusion matrix : \n',confusion_matrix(y_test,df_knn_pred_test))
sns.heatmap(confusion_matrix(y_test,df_knn_pred_test), annot = True)
plt.show()
```

```
Classification report :
               precision    recall  f1-score   support

           0       0.51      0.54      0.53        39
           1       0.55      0.52      0.54        42

    accuracy                           0.53        81
   macro avg       0.53      0.53      0.53        81
weighted avg       0.53      0.53      0.53        81

confusion matrix :
 [[21 18]
 [20 22]]
```

The above fig. shows the analysis of classification report of Random Forrest using confusion matrix where the **Accuracy is 0.53**

The evaluation metrics of various machine learning models are evaluated. From the analysis, the heart disease (Presence or Absence) for the dataset considered can be better predicted using the Random Forrest Algorithm. Therefore, the Majority Class for the dataset can be determine using **Random Forrest Algorithm** is **84%**

This dataset can be considered to be a Positive-class baseline since the value of absence of heart disease is much higher than the value of presence of heart disease. The rate of **Presence** of heart disease is **120 of 270** whereas the rate of **Absence** of heart disease is **150 of 270**. Frrom the dataset, the percentage of **Absence** of heart disease is **55.6%** and the percentage of **Presence** of heart disease is **44.4%.**

Proposed Methodology:

The supervised learning is used to determine the efficiency of algorithms for the dataset. Here, Logistic Regression, Decision Tree, Random Forrest and KNN algorithm are used as classifiers to classify the data. The data is classified into training data and testing data as it helps in classification analysis. The target variable of this dataset is heart disease (Presence or Absence). To determine the efficiency, first data needs to be split into training and testing data where variables such as Age, sex, Max HR, BP, cholesterol are used as input to the data and the target variable is given as heart disease (Presence and Absence). After data splitting, normalisation is done by means of feature scaling. Then data is engineered to find the suitable features and to explore news ones for better analysis. Then the data is evaluated using the machine learning techniques to find the Accuracy of training and testing data and also to evaluate F1 score of the data. Finally, classification report is evaluated for each algorithm to determine the best fir for prediction of heart disease. This project can be further enhanced by using these algorithms to evaluate the metrics such as hyperparameters, recall, using machine learning algorithms