```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
from sklearn.metrics import log_loss,roc_auc_score,precision_score,f1_score,recall_score,r
from sklearn.metrics import classification_report, confusion_matrix,accuracy_score,fbeta_s
from sklearn import metrics
from sklearn.model_selection import train_test_split
# machine learning algorithms
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier,VotingClassifier,AdaBoostClassifier,Gr
from sklearn.neural_network import MLPClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import SGDClassifier
from sklearn.svm import SVC
```

```
df = pd.read_csv('heart-statlog.csv')
df.head()
```

	age	sex	chest	resting_blood_pressure	serum_cholestoral	fasting_blood_sugar
0	70	1	4	130	322	0
1	67	0	3	115	564	0
2	57	1	2	124	261	0
3	64	1	4	128	263	0
4	74	0	2	120	269	0

```
df["class"].replace({"present": 1, "absent": 0}, inplace=True)
```

```
# segregating dataset into features i.e., X and target variables i.e., y
X = df.drop(columns=['class'])
y = df['class']

X_train, X_test, y_train, y_test = train_test_split(X, y, stratify=y, test_size=0.2,shuffl
```

```
df.head()
```

	age	sex	chest	resting_blood_pressure	serum_cholestoral	fasting_blood_sugar
0	70	1	4	130	322	0

```
## checking distribution of traget variable in train test split
print('Distribution of traget variable in training set')
print(y_train.value_counts())

print('Distribution of traget variable in test set')
print(y_test.value_counts())
```

```
Distribution of traget variable in training set
0    120
1     96
Name: class, dtype: int64
Distribution of traget variable in test set
0    30
1    24
Name: class, dtype: int64
```

```
print('------------Training Set------------------')
print(X_train.shape)
print(y_train.shape)

print('------------Test Set------------------')
print(X_test.shape)
print(y_test.shape)
```

```
------------Training Set------------------
(216, 13)
(216,)
------------Test Set------------------
(54, 13)
(54,)
```

```
 from sklearn import model_selection

rf_ent = RandomForestClassifier(criterion='entropy',n_estimators=270)
rf_ent.fit(X_train, y_train)
y_pred_rfe = rf_ent.predict(X_test)

scoring = 'accuracy'
results = []
kfold = model_selection.KFold(n_splits=10)
cv_results = model_selection.cross_val_score(RandomForestClassifier(criterion='entropy',n_
results.append(cv_results)
#names.append(name)
msg = "%s: %f (%f)" % ('RF_Ent100', cv_results.mean(), cv_results.std())
print(msg)
```

```
RF_Ent100: 0.841991 (0.056410)
```

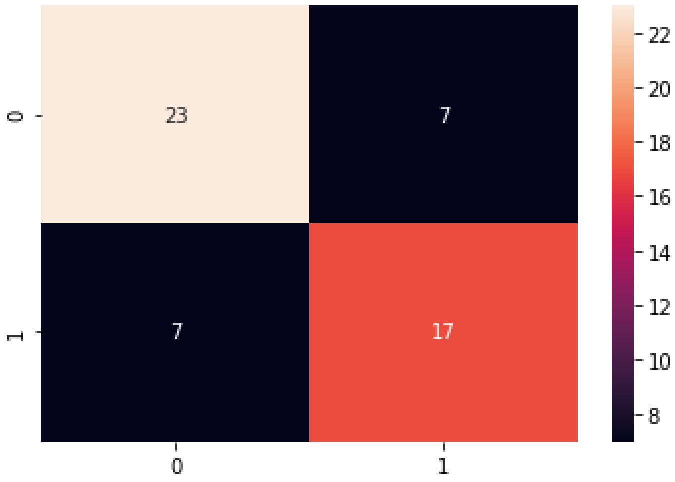
```
CM=confusion_matrix(y_test,y_pred_rfe)
sns.heatmap(CM, annot=True)
```

```
TN = CM[0][0]
FN = CM[1][0]
TP = CM[1][1]
FP = CM[0][1]
specificity = TN/(TN+FP)
sensitivity = TP/(TP+FN)
loss_log = log_loss(y_test, y_pred_rfe)
acc= accuracy_score(y_test, y_pred_rfe)
roc=roc_auc_score(y_test, y_pred_rfe)
prec = precision_score(y_test, y_pred_rfe)
rec = recall_score(y_test, y_pred_rfe)
f1 = f1_score(y_test, y_pred_rfe)

model_results =pd.DataFrame([['Random Forest', acc, prec, rec, f1, sensitivity, specificit
               columns = ['Model','Accuracy','Precision','Recall','f1 score','Sensitivity'

model_results
```

	Model	Accuracy	Precision	Recall	f1 score	Sensitivity	Specificity
0	Random Forest	0.740741	0.708333	0.708333	0.708333	0.708333	0.766667



```
pip install sklearn-genetic
```

```
Collecting sklearn-genetic
  Downloading sklearn_genetic-0.4.1-py2.py3-none-any.whl (10 kB)
Collecting deap>=1.0.2
  Downloading deap-1.3.1-cp37-cp37m-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_
     |████████████████████████████████| 160 kB 5.2 MB/s
Requirement already satisfied: scikit-learn>=0.20.3 in /usr/local/lib/python3.7/dist-
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: scipy>=0.17.0 in /usr/local/lib/python3.7/dist-package
Installing collected packages: deap, sklearn-genetic
Successfully installed deap-1.3.1 sklearn-genetic-0.4.1
```

```
from genetic_selection import GeneticSelectionCV

# import your preferred ml model.
from sklearn.ensemble import RandomForestClassifier
```

```
#build the model with your preferred hyperparameters.
rf_ent = RandomForestClassifier(n_estimators=270)

# create the GeneticSelection search with the different parameters available.
rf_ent = GeneticSelectionCV(rf_ent,
                              cv=10,
                              scoring="accuracy",
                              max_features=13,
                              n_population=270,
                              crossover_proba=0.5,
                              mutation_proba=0.2,
                              n_generations=50,
                              crossover_independent_proba=0.5,
                              mutation_independent_proba=0.05,
                              n_gen_no_change=10,
                              n_jobs=-1)

# fit the GA search to our data.
rf_ent = rf_ent.fit(X_train, y_train)

# print the results.
print(rf_ent.support_)
```

```
     [ True  True  True False False False False False  True  True  True  True
       True]
```

```
y_pred_rfe = rf_ent.predict(X_test)
print("Accuracy score after genetic algorithm is= "+str(accuracy_score(y_test,y_pred_rfe))
```

```
     Accuracy score after genetic algorithm is= 0.8333333333333334
```

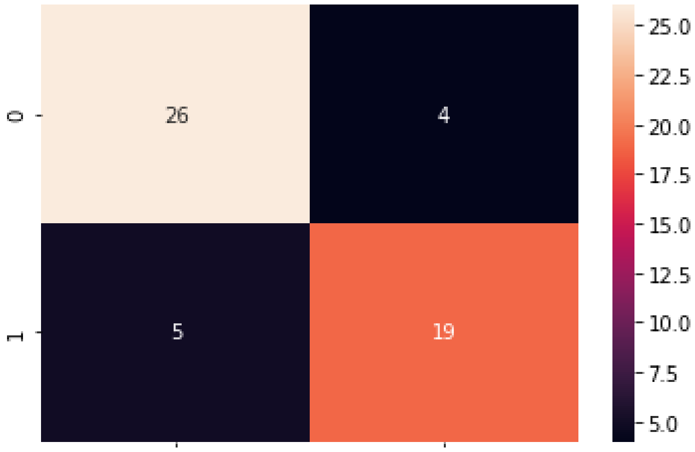
```
CM=confusion_matrix(y_test,y_pred_rfe)
sns.heatmap(CM, annot=True)

TN = CM[0][0]
FN = CM[1][0]
TP = CM[1][1]
FP = CM[0][1]
specificity = TN/(TN+FP)
sensitivity = TP/(TP+FN)
loss_log = log_loss(y_test, y_pred_rfe)
acc= accuracy_score(y_test, y_pred_rfe)
roc=roc_auc_score(y_test, y_pred_rfe)
prec = precision_score(y_test, y_pred_rfe)
rec = recall_score(y_test, y_pred_rfe)
f1 = f1_score(y_test, y_pred_rfe)

model_results =pd.DataFrame([['Random Forest', acc, prec, rec, f1, sensitivity, specificit
               columns = ['Model','Accuracy','Precision','Recall','f1 score','Sensitivity'

model_results
```

	Model	Accuracy	Precision	Recall	f1 score	Sensitivity	Specificity	
0	Random Forest	0.833333	0.826087	0.791667	0.808511	0.791667	0.866667	(


26
4
5
19
0
1
25.0
22.5
20.0
17.5
15.0
12.5
10.0
7.5
5.0