confusion-matrix

October 16, 2023

[1]:
```
#Loading libraries
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
import seaborn as sb
```

[2]:
```
#Import data
df = pd.read_csv('insurance.csv')
df.head()
```

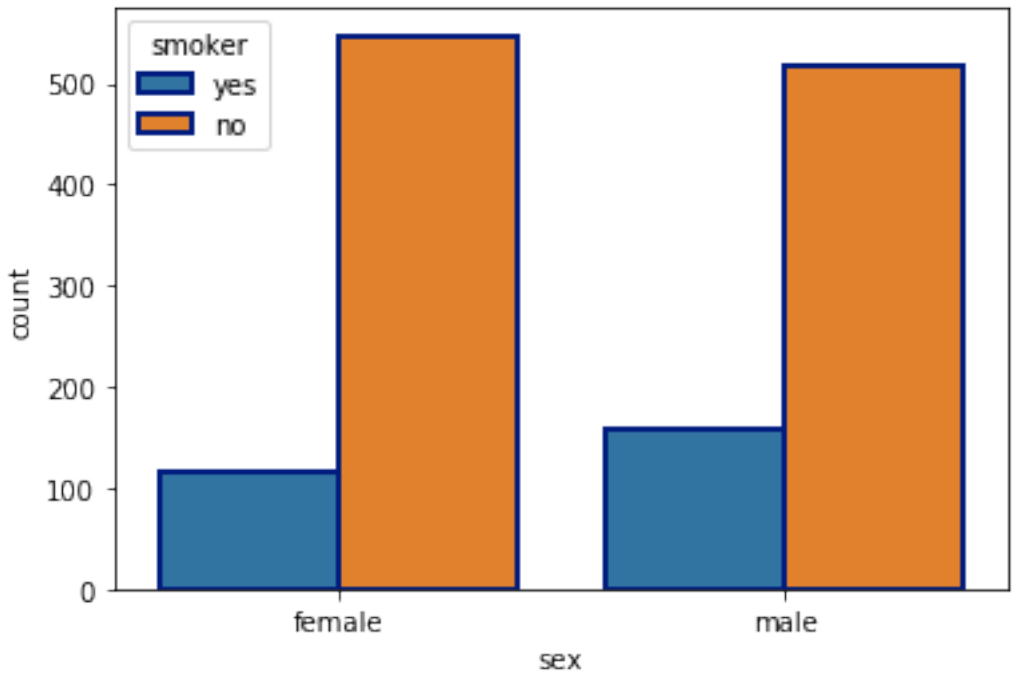
[2]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520

0.0.1 Data Correlation

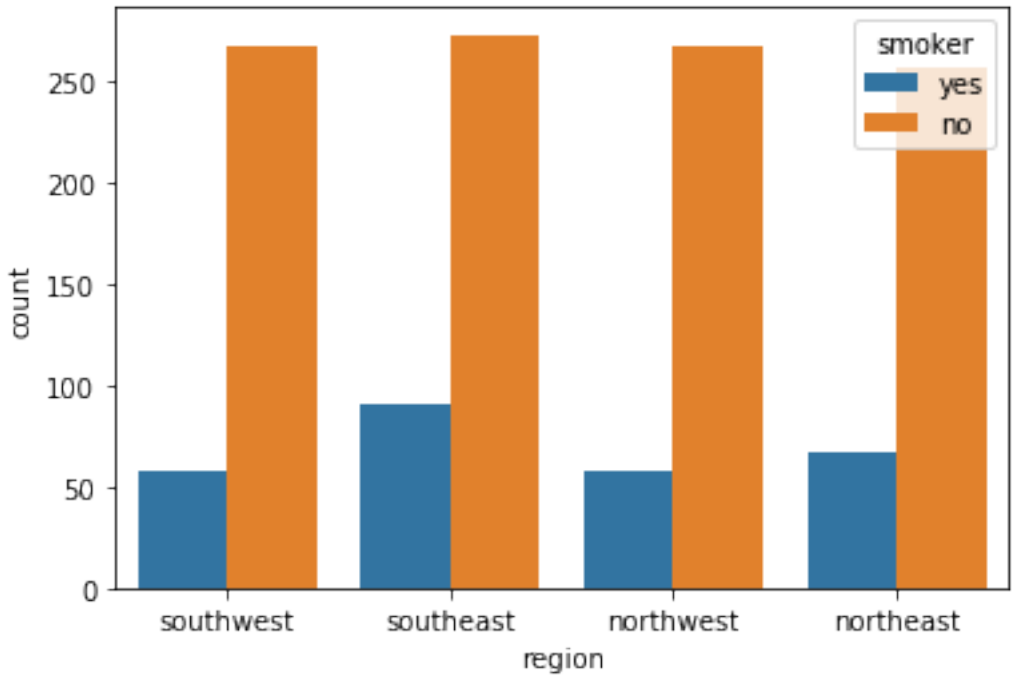
[3]:
```
#sex based total smoker
sb.countplot(x='sex',data=df,hue='smoker',linewidth=2,edgecolor=sb.
 ↪color_palette("dark", 1))
```

[3]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x24ab7ed9088>
```



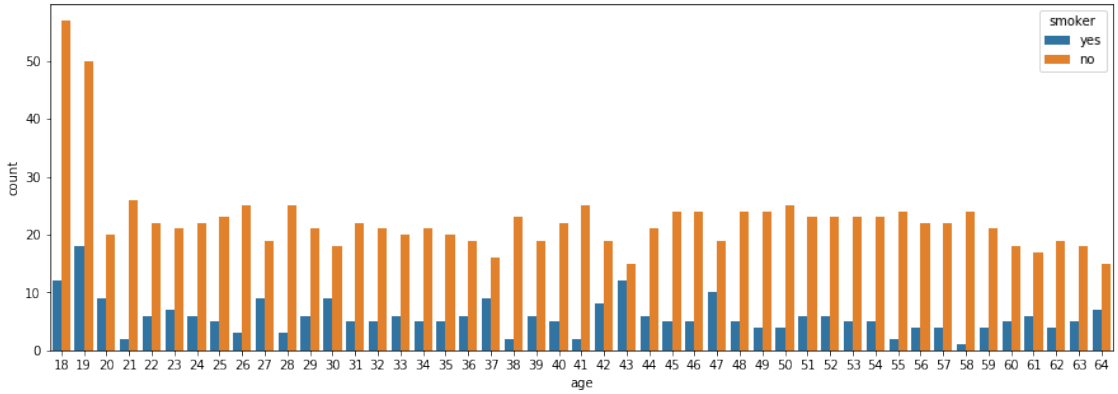
[4]:
```
#area based total smoker
sb.countplot(data=df, x='region',hue='smoker')
```

[4]: <matplotlib.axes._subplots.AxesSubplot at 0x24ab7fe7048>



[5]:
```
#age based total smoker
plt.figure(figsize=(15,5))
sb.countplot(data=df, x='age',hue='smoker')
```

[5]: <matplotlib.axes._subplots.AxesSubplot at 0x24ab806b548>



0.0.2 Encoding the dataframe

[6]:
```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
```

[7]:
```
#Encoding all column in a single shot using pandas
from pandas.core.dtypes.common import is_numeric_dtype
for column in df.columns:
    if is_numeric_dtype(df[column]):
        continue
    df[column] = le.fit_transform(df[column])
```

[8]:
```
df.head()
```

[8]:

	age	sex	bmi	children	smoker	region	charges
0	19	0	27.900	0	1	3	16884.92400
1	18	1	33.770	1	0	2	1725.55230
2	28	1	33.000	3	0	2	4449.46200
3	33	1	22.705	0	0	1	21984.47061
4	32	1	28.880	0	0	1	3866.85520

[9]:
```
#summary of dataframe
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1338 entries, 0 to 1337
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       1338 non-null   int64
 1   sex       1338 non-null   int32
 2   bmi       1338 non-null   float64
 3   children  1338 non-null   int64
 4   smoker    1338 non-null   int32
 5   region    1338 non-null   int32
 6   charges   1338 non-null   float64
dtypes: float64(2), int32(3), int64(2)
memory usage: 57.6 KB
```

[10]:
```
#unique values of a column
df.smoker.unique()
```

[10]:
```
array([1, 0])
```

[11]:
```
#count value type
df.smoker.value_counts()
```

[11]:
```
0    1064
1     274
Name: smoker, dtype: int64
```

[12]:
```
#smoker percentage
smoker_percentage = (274/(1064+274))*100
smoker_percentage
```

[12]:
```
20.47832585949178
```

[13]:
```
#Drop all rows with null values
#df.dropna(inplace=True)

x = df.drop('smoker',axis=1)
y = df['smoker']
```

0.1 Decision Tree

[14]:
```
from sklearn.tree import DecisionTreeClassifier
dtc = DecisionTreeClassifier()
```

[15]:
```
dtc.fit(x,y)
```

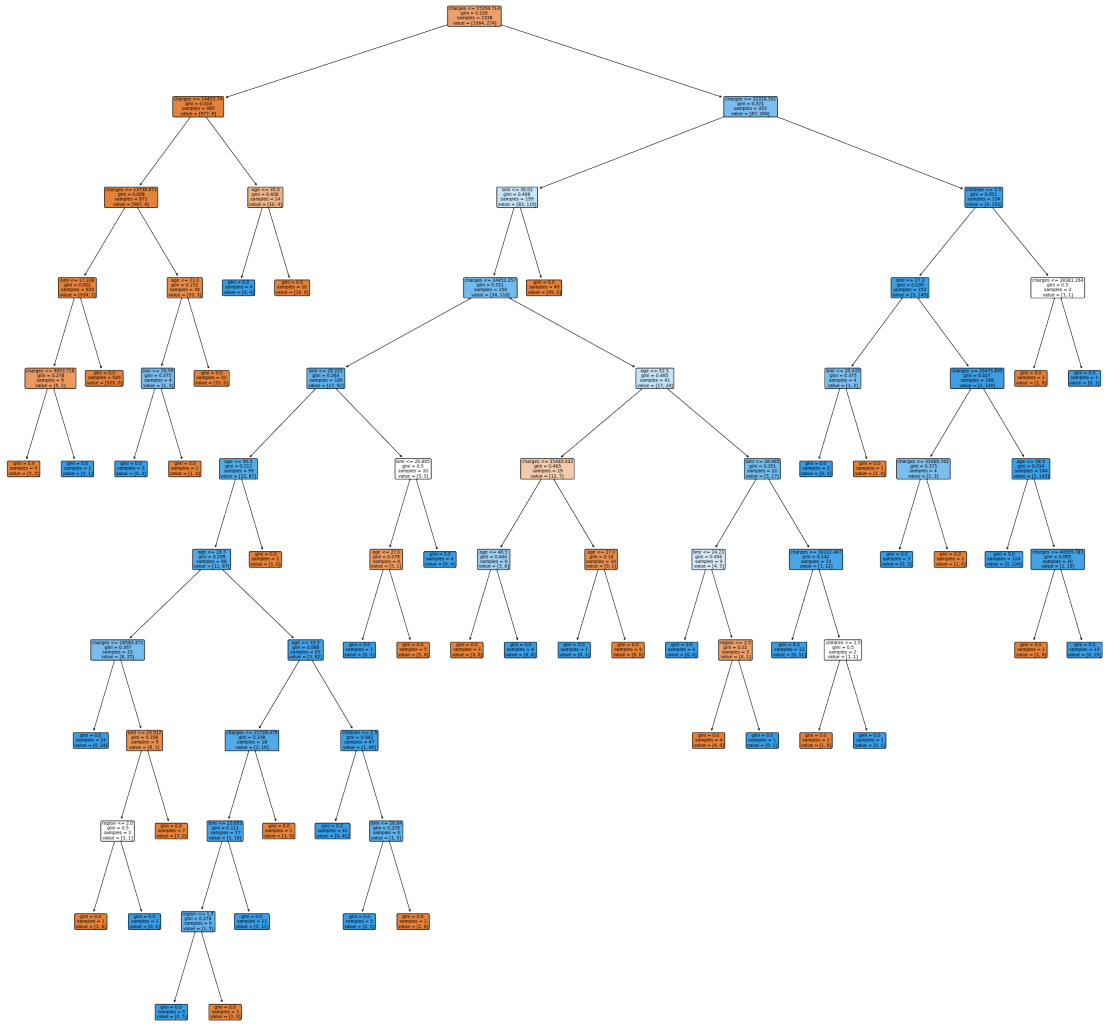
[15]:
```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
                       max_depth=None, max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort='deprecated',
                       random_state=None, splitter='best')
```

[16]:
```
from sklearn import tree

#set figure size
plt.figure(figsize=(30,30))

#plot decision tree
tree.plot_tree(dtc,filled=True,rounded = True,feature_names = x.columns)

#save current figure
plt.savefig("decision_tree.png")
```



[17]:
```
#plotting tree in stream
print(tree.export_text(dtc,show_weights = True,feature_names=list(x.columns)))
```

```
|--- charges <= 15294.71
|   |--- charges <= 14453.74
|   |   |--- charges <= 13736.67
|   |   |   |--- bmi <= 17.34
|   |   |   |   |--- charges <= 9853.72
|   |   |   |   |   |--- weights: [5.00, 0.00] class: 0
|   |   |   |   |--- charges >  9853.72
|   |   |   |   |   |--- weights: [0.00, 1.00] class: 1
|   |   |   |--- bmi >  17.34
|   |   |   |   |--- weights: [929.00, 0.00] class: 0
|   |   |--- charges >  13736.67
```

```
|   |   |   |--- age <= 21.00
|   |   |   |   |--- bmi <= 29.99
|   |   |   |   |   |--- weights: [0.00, 3.00] class: 1
|   |   |   |   |--- bmi >  29.99
|   |   |   |   |   |--- weights: [1.00, 0.00] class: 0
|   |   |   |--- age >  21.00
|   |   |   |   |--- weights: [32.00, 0.00] class: 0
|   |--- charges >  14453.74
|   |   |--- age <= 35.00
|   |   |   |--- weights: [0.00, 4.00] class: 1
|   |   |--- age >  35.00
|   |   |   |--- weights: [10.00, 0.00] class: 0
|--- charges >  15294.71
|   |--- charges <= 32328.50
|   |   |--- bmi <= 30.01
|   |   |   |--- charges <= 24453.36
|   |   |   |   |--- bmi <= 29.22
|   |   |   |   |   |--- age <= 56.50
|   |   |   |   |   |   |--- age <= 28.50
|   |   |   |   |   |   |   |--- charges <= 18583.47
|   |   |   |   |   |   |   |   |--- weights: [0.00, 24.00] class: 1
|   |   |   |   |   |   |   |--- charges >  18583.47
|   |   |   |   |   |   |   |   |--- bmi <= 23.01
|   |   |   |   |   |   |   |   |   |--- region <= 2.00
|   |   |   |   |   |   |   |   |   |   |--- weights: [1.00, 0.00] class: 0
|   |   |   |   |   |   |   |   |   |--- region >  2.00
|   |   |   |   |   |   |   |   |   |   |--- weights: [0.00, 1.00] class: 1
|   |   |   |   |   |   |   |   |--- bmi >  23.01
|   |   |   |   |   |   |   |   |   |--- weights: [7.00, 0.00] class: 0
|   |   |   |   |   |   |--- age >  28.50
|   |   |   |   |   |   |   |--- age <= 33.50
|   |   |   |   |   |   |   |   |--- charges <= 21728.47
|   |   |   |   |   |   |   |   |   |--- bmi <= 23.65
|   |   |   |   |   |   |   |   |   |   |--- region <= 1.50
|   |   |   |   |   |   |   |   |   |   |   |--- weights: [0.00, 5.00] class: 1
|   |   |   |   |   |   |   |   |   |   |--- region >  1.50
|   |   |   |   |   |   |   |   |   |   |   |--- weights: [1.00, 0.00] class: 0
|   |   |   |   |   |   |   |   |   |--- bmi >  23.65
|   |   |   |   |   |   |   |   |   |   |--- weights: [0.00, 11.00] class: 1
|   |   |   |   |   |   |   |   |--- charges >  21728.47
|   |   |   |   |   |   |   |   |   |--- weights: [1.00, 0.00] class: 0
|   |   |   |   |   |   |--- age >  33.50
|   |   |   |   |   |   |   |--- children <= 2.50
|   |   |   |   |   |   |   |   |--- weights: [0.00, 41.00] class: 1
|   |   |   |   |   |   |   |--- children >  2.50
|   |   |   |   |   |   |   |   |--- bmi <= 26.84
|   |   |   |   |   |   |   |   |   |--- weights: [0.00, 5.00] class: 1
|   |   |   |   |   |   |   |   |--- bmi >  26.84
```

```
|   |   |   |   |   |   |   |   |   |   |--- weights: [1.00, 0.00] class: 0
|   |   |   |   |   |   |--- age >  56.50
|   |   |   |   |   |   |   |--- weights: [1.00, 0.00] class: 0
|   |   |   |   |   |--- bmi >  29.22
|   |   |   |   |   |   |--- bmi <= 29.80
|   |   |   |   |   |   |   |--- age <= 27.00
|   |   |   |   |   |   |   |   |--- weights: [0.00, 1.00] class: 1
|   |   |   |   |   |   |   |--- age >  27.00
|   |   |   |   |   |   |   |   |--- weights: [5.00, 0.00] class: 0
|   |   |   |   |   |   |--- bmi >  29.80
|   |   |   |   |   |   |   |--- weights: [0.00, 4.00] class: 1
|   |   |   |   |--- charges >  24453.36
|   |   |   |   |   |--- age <= 52.50
|   |   |   |   |   |   |--- charges <= 25483.03
|   |   |   |   |   |   |   |--- age <= 46.50
|   |   |   |   |   |   |   |   |--- weights: [3.00, 0.00] class: 0
|   |   |   |   |   |   |   |--- age >  46.50
|   |   |   |   |   |   |   |   |--- weights: [0.00, 6.00] class: 1
|   |   |   |   |   |   |--- charges >  25483.03
|   |   |   |   |   |   |   |--- age <= 27.00
|   |   |   |   |   |   |   |   |--- weights: [0.00, 1.00] class: 1
|   |   |   |   |   |   |   |--- age >  27.00
|   |   |   |   |   |   |   |   |--- weights: [9.00, 0.00] class: 0
|   |   |   |   |   |--- age >  52.50
|   |   |   |   |   |   |--- bmi <= 26.07
|   |   |   |   |   |   |   |--- bmi <= 24.23
|   |   |   |   |   |   |   |   |--- weights: [0.00, 4.00] class: 1
|   |   |   |   |   |   |   |--- bmi >  24.23
|   |   |   |   |   |   |   |   |--- region <= 2.50
|   |   |   |   |   |   |   |   |   |--- weights: [4.00, 0.00] class: 0
|   |   |   |   |   |   |   |   |--- region >  2.50
|   |   |   |   |   |   |   |   |   |--- weights: [0.00, 1.00] class: 1
|   |   |   |   |   |--- bmi >  26.07
|   |   |   |   |   |   |--- charges <= 30222.47
|   |   |   |   |   |   |   |--- weights: [0.00, 11.00] class: 1
|   |   |   |   |   |   |--- charges >  30222.47
|   |   |   |   |   |   |   |--- children <= 1.50
|   |   |   |   |   |   |   |   |--- weights: [1.00, 0.00] class: 0
|   |   |   |   |   |   |   |--- children >  1.50
|   |   |   |   |   |   |   |   |--- weights: [0.00, 1.00] class: 1
|   |   |--- bmi >  30.01
|   |   |   |--- weights: [49.00, 0.00] class: 0
|   |--- charges >  32328.50
|   |   |--- children <= 3.50
|   |   |   |--- bmi <= 27.30
|   |   |   |   |--- bmi <= 26.43
|   |   |   |   |   |--- weights: [0.00, 3.00] class: 1
|   |   |   |   |--- bmi >  26.43
```

```
|   |   |   |   |   |--- weights: [1.00, 0.00] class: 0
|   |   |   |--- bmi >  27.30
|   |   |   |   |--- charges <= 33473.89
|   |   |   |   |   |--- charges <= 33389.76
|   |   |   |   |   |   |--- weights: [0.00, 3.00] class: 1
|   |   |   |   |   |--- charges >  33389.76
|   |   |   |   |   |   |--- weights: [1.00, 0.00] class: 0
|   |   |   |   |--- charges >  33473.89
|   |   |   |   |   |--- age <= 58.50
|   |   |   |   |   |   |--- weights: [0.00, 124.00] class: 1
|   |   |   |   |   |--- age >  58.50
|   |   |   |   |   |   |--- charges <= 40959.78
|   |   |   |   |   |   |   |--- weights: [1.00, 0.00] class: 0
|   |   |   |   |   |   |--- charges >  40959.78
|   |   |   |   |   |   |   |--- weights: [0.00, 19.00] class: 1
|   |   |--- children >  3.50
|   |   |   |--- charges <= 38381.26
|   |   |   |   |--- weights: [1.00, 0.00] class: 0
|   |   |   |--- charges >  38381.26
|   |   |   |   |--- weights: [0.00, 1.00] class: 1
```

[18]:
```
from sklearn.model_selection import train_test_split

#Split into random train and test subsets
xtrain,xtest,ytrain,ytest = train_test_split(x,y,test_size=.3,random_state=42)
```

[19]:
```
xtrain.shape
```

[19]: (936, 6)

[20]:
```
xtest.shape
```

[20]: (402, 6)

[21]:
```
y.index
```

[21]: RangeIndex(start=0, stop=1338, step=1)

[22]:
```
#independent features
x.columns
```

[22]: Index(['age', 'sex', 'bmi', 'children', 'region', 'charges'], dtype='object')

[23]:
```
#dependent feature - on x
y.head()
```

[23]:
```
0    1
1    0
2    0
3    0
4    0
Name: smoker, dtype: int32
```

[24]:
```
dtc.fit(xtrain,ytrain)
```

[24]:
```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
                       max_depth=None, max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort='deprecated',
                       random_state=None, splitter='best')
```

[25]:
```
#predict for testing dataset
pred = dtc.predict(xtest)
```

[26]:
```
#predicted values for testing data
pred
```

[26]:
```
array([0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
       1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0,
       0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1,
       0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1,
       0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1,
       1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1,
       0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0,
       0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1,
       0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0,
       0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
       0, 0, 0, 0, 0, 0])
```

[27]:
```
#actual values for testing data
np.array(ytest)
```

[27]:
```
array([0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0,
```

```
0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0,
0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1,
0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1,
0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1,
1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1,
0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0,
0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1,
0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0,
0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0])
```

0.2 Performance Measurement (on Classification)

0.2.1 Confusion Matrix

[28]:
```
from sklearn.metrics import confusion_matrix
cmatrix = confusion_matrix(ytest,pred)
cmatrix
```

[28]:
```
array([[313,  10],
       [  3,  76]], dtype=int64)
```

[29]:
```
tp,fn,fp,tn = cmatrix.reshape(-1)
```

[30]:
```
tp
```

[30]: 313

[31]:
```
fn
```

[31]: 10

[32]:
```
fp
```

[32]: 3

[33]:
```
tn
```

[33]: 76

0.2.2 Accuracy

[34]:
```
from sklearn.metrics import classification_report,plot_roc_curve,accuracy_score
```

[35]:
```
(tp+tn)/len(ytest)
```

[35]: 0.9676616915422885

[36]:
```
dtc.score(xtest,ytest)
```

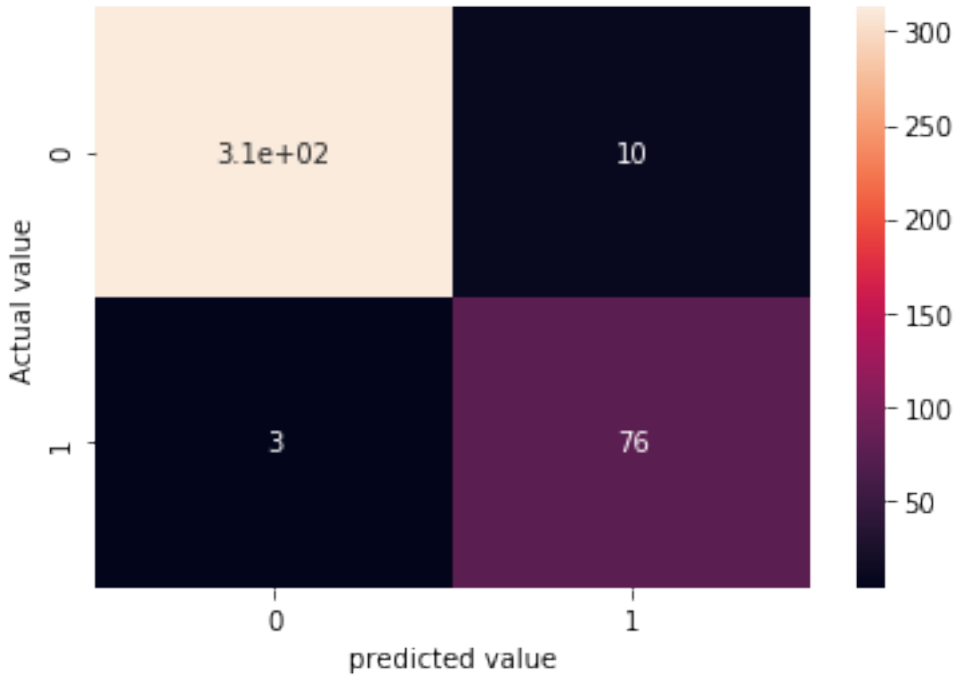
[36]: 0.9676616915422885

[37]:
```
accuracy_score(ytest,pred)
```

[37]: 0.9676616915422885

[38]:
```
sb.heatmap(cmatrix,annot=True)
plt.xlabel('predicted value')
plt.ylabel('Actual value')
```

[38]: Text(33.0, 0.5, 'Actual value')



0.2.3 Classification Report

[39]:
```
print(classification_report(ytest,pred))
```

```
              precision    recall  f1-score   support

           0       0.99      0.97      0.98       323
           1       0.88      0.96      0.92        79

    accuracy                           0.97       402
   macro avg       0.94      0.97      0.95       402
weighted avg       0.97      0.97      0.97       402
```

0.2.4 Precision / Positive Predictive Value (PPV)

[40]:
```
ppv = tp / (tp+fp)
ppv
```

[40]: 0.990506329113924

0.2.5 Sensitivity / Recall / Hit Rate / True Positive Rate (TPR)

[41]:
```
tpr = tp / (tp+fn)
tpr
```

[41]: 0.9690402476780186

0.2.6 False Positive Rate(FPR) / 1-Specificity

[42]:
```
fp/(fp+tn)
```

[42]: 0.0379746835443038

0.2.7 F1 Measure

[43]:
```
(ppv + tpr) / 2
```

[43]: 0.9797732883959713

0.2.8 F1 score / Harmonic Mean

[44]:
```
(2 * ppv * tpr) / (ppv + tpr)
```

[44]: 0.9796557120500782

0.2.9 Specificity / Selectivity / True Negative Rate (TNR)

[45]:
```
tnr = tn / (tn + fp)
tnr
```

[45]: 0.9620253164556962

0.2.10 Threat Score / Critical Success Index (CSI)

[46]:
```
tp / (tp + fp + fn)
```

[46]: 0.9601226993865031

0.2.11 False Discovery Rate (FDR)

[47]:
```
fp / (tp + fp)
```

[47]: 0.00949367088607595

0.2.12 Balanced Accuracy (BA)

[48]:
```
(tpr + tnr) / 2
```

[48]: 0.9655327820668573

0.2.13 Informadness / Bookmaker Informadness (BM)

[49]:
```
tpr +tnr -1
```

[49]: 0.9310655641337147

0.2.14 Negative Predictive Value (NPV)

[50]:
```
npv = tn / (tn + fn)
npv
```

[50]: 0.8837209302325582

0.2.15 Markedness (MK) / delta-p

[51]:
```
ppv + npv -1
```

[51]: 0.8742272593464822

0.2.16 AuC(Area Under Curve) & RoC(Receiver Operating Characteristics) graph

[52]:
```
plot_roc_curve(dtc,xtest,ytest)
plt.plot([0,1],[0,1])
```

[52]:
```
[<matplotlib.lines.Line2D at 0x24ab9b24a88>]
```