In [1]:
```
#1)decision tree
#2)random forest
#3)logistic regression
#4)KNN
#5)SVM
#assignmemnt-06
#D.prudhvi sai
```

In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [3]:
```
dataset=pd.read_csv('diabetes.csv')
```

In [4]:
```
dataset.isnull().any()
```

Out[4]:
```
Pregnancies                 False
Glucose                     False
BloodPressure               False
SkinThickness               False
Insulin                     False
BMI                         False
DiabetesPedigreeFunction    False
Age                         False
Diabetes                    False
dtype: bool
```

In [5]:
```
dataset
```

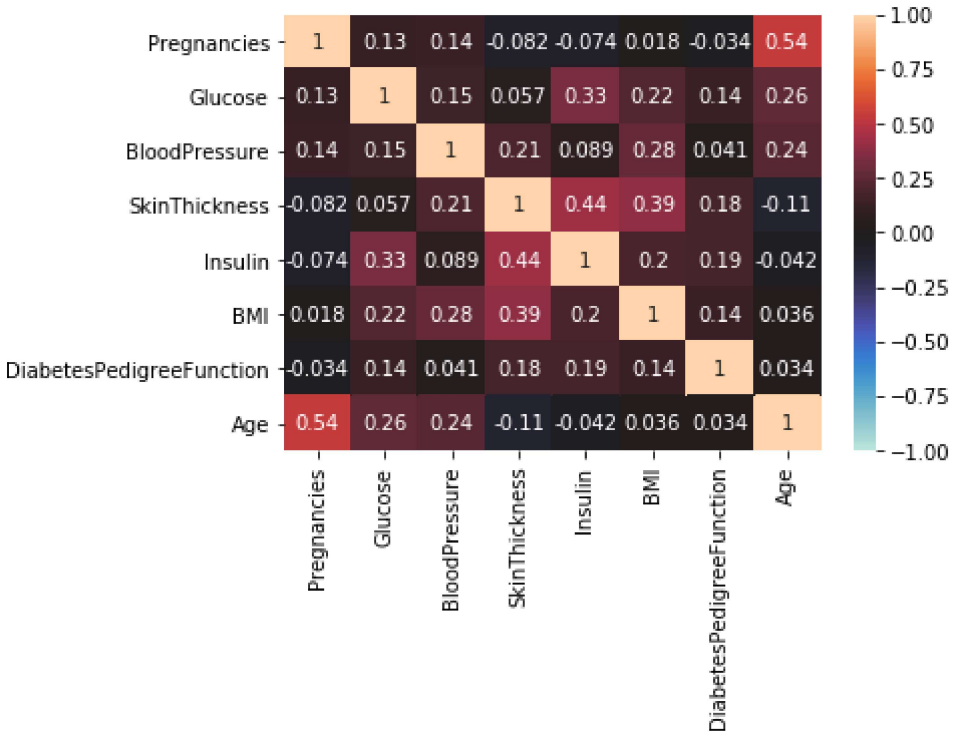
Out[5]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction
0	6	148	72	35	0	33.6	0.627
1	1	85	66	29	0	26.6	0.351
2	8	183	64	0	0	23.3	0.672
3	1	89	66	23	94	28.1	0.167
4	0	137	40	35	168	43.1	2.288
...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171
764	2	122	70	27	0	36.8	0.340
765	5	121	72	23	112	26.2	0.245
766	1	126	60	0	0	30.1	0.349
767	1	93	70	31	0	30.4	0.315

768 rows × 9 columns

In [6]:
```
import seaborn as sns
sns.heatmap(dataset.corr(),annot=True,vmin=-1,vmax=1,center=0)
```

Out[6]: <matplotlib.axes._subplots.AxesSubplot at 0x29713d42888>



In [7]:
```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
```

In [8]:
```
dataset['Diabetes']=le.fit_transform(dataset['Diabetes'])
```

In [9]:
```
dataset
```

Out[9]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction
0	6	148	72	35	0	33.6	0.627
1	1	85	66	29	0	26.6	0.351
2	8	183	64	0	0	23.3	0.672
3	1	89	66	23	94	28.1	0.167
4	0	137	40	35	168	43.1	2.288
...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171
764	2	122	70	27	0	36.8	0.340
765	5	121	72	23	112	26.2	0.245
766	1	126	60	0	0	30.1	0.349
767	1	93	70	31	0	30.4	0.315

768 rows × 9 columns

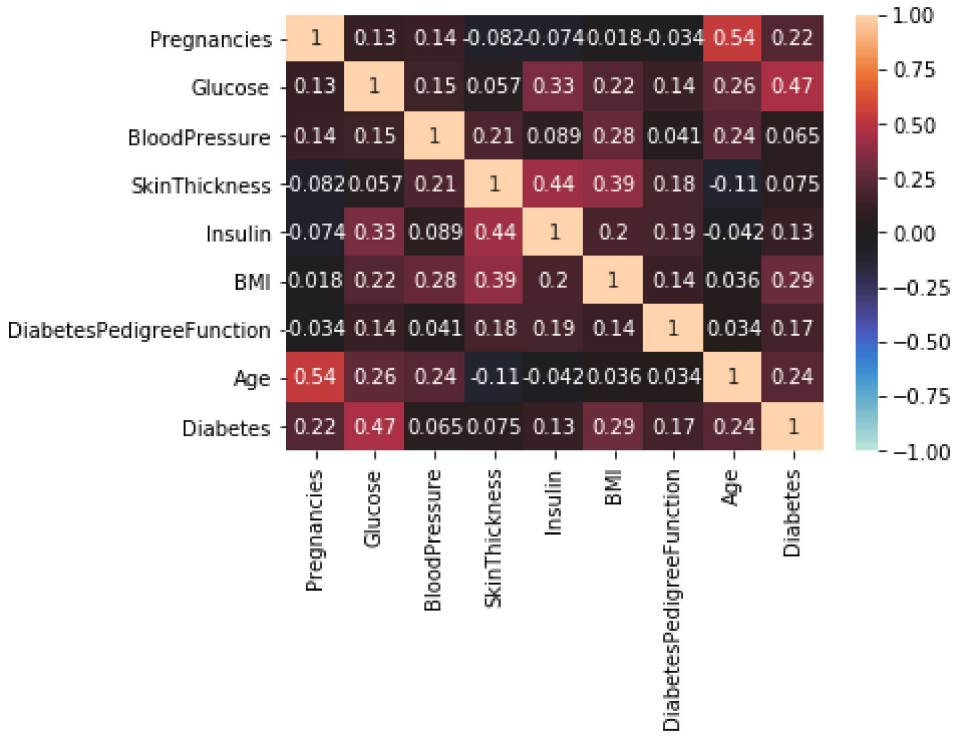
In [10]:
```
dataset.corr()
```

Out[10]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BI
Pregnancies	1.000000	0.129459	0.141282	-0.081672	-0.073535	0.01768
Glucose	0.129459	1.000000	0.152590	0.057328	0.331357	0.22107
BloodPressure	0.141282	0.152590	1.000000	0.207371	0.088933	0.28180
SkinThickness	-0.081672	0.057328	0.207371	1.000000	0.436783	0.39257
Insulin	-0.073535	0.331357	0.088933	0.436783	1.000000	0.19785
BMI	0.017683	0.221071	0.281805	0.392573	0.197859	1.00000
DiabetesPedigreeFunction	-0.033523	0.137337	0.041265	0.183928	0.185071	0.14064
Age	0.544341	0.263514	0.239528	-0.113970	-0.042163	0.03624
Diabetes	0.221898	0.466581	0.065068	0.074752	0.130548	0.29269

In [11]:
```
import seaborn as sns
sns.heatmap(dataset.corr(),annot=True,vmin=-1,vmax=1,center=0)
```

Out[11]: <matplotlib.axes._subplots.AxesSubplot at 0x297146cb1c8>



In [12]:
```
x=dataset.iloc[:,0:8]
```

In [13]:
```
x.head()
```

Out[13]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	A
0	6	148	72	35	0	33.6	0.627	
1	1	85	66	29	0	26.6	0.351	
2	8	183	64	0	0	23.3	0.672	
3	1	89	66	23	94	28.1	0.167	
4	0	137	40	35	168	43.1	2.288	

In [14]:
```
y=dataset.iloc[:,8]
```

In [15]:
```
y.head()
```

Out[15]:
```
0    1
1    0
2    1
3    0
4    1
Name: Diabetes, dtype: int32
```

In [16]:
```
x=dataset.iloc[:,0:8].values
y=dataset.iloc[:,-1:].values
```

In [17]:
```
x
```

Out[17]:
```
array([[  6.   , 148.   ,  72.   , ...,  33.6  ,   0.627,  50.   ],
       [  1.   ,  85.   ,  66.   , ...,  26.6  ,   0.351,  31.   ],
       [  8.   , 183.   ,  64.   , ...,  23.3  ,   0.672,  32.   ],
       ...,
       [  5.   , 121.   ,  72.   , ...,  26.2  ,   0.245,  30.   ],
       [  1.   , 126.   ,  60.   , ...,  30.1  ,   0.349,  47.   ],
       [  1.   ,  93.   ,  70.   , ...,  30.4  ,   0.315,  23.   ]])
```

In [18]:
```
y
```

Out[18]:
```
       [1],
       [1],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [1],
       [1],
       [0],
       [0],
       [1],
       [0],
       [0],
       [1],
       [0],
       [1],
       [1],
       [1],
```

In [19]:
```
from sklearn.model_selection import train_test_split
```

In [106]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2, random_state=6)
```

In [107]:
```
x_train.shape
```

Out[107]:
```
(614, 8)
```

In [108]:
```
y_train.shape
```

Out[108]:
```
(614, 1)
```

In [109]:
```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.fit_transform(x_test)
```

In [128]:
```
#decision tree
from sklearn.tree import DecisionTreeClassifier
dtc=DecisionTreeClassifier(criterion='entropy')
dtc.fit(x_train,y_train)
```

Out[128]:
```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='entropy',
                       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')
```

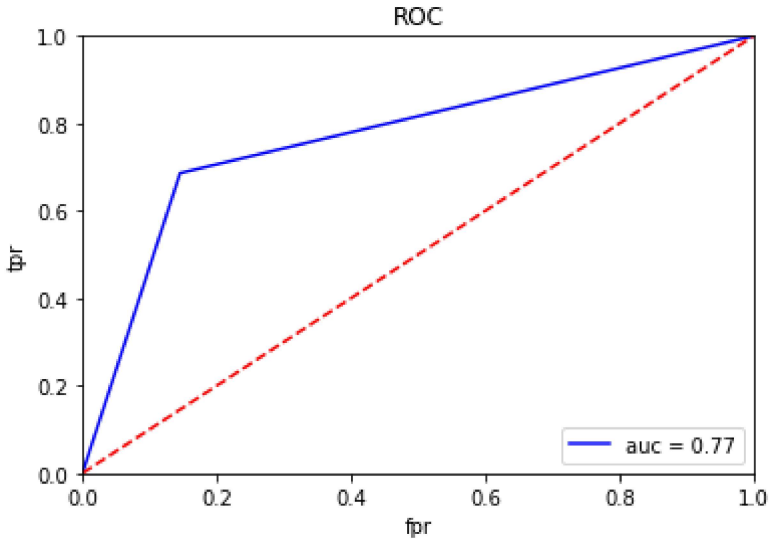
In [129]:
```
y_pred = dtc.predict(x_test)
y_pred
```

Out[129]:
```
array([0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0,
       0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0,
       0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0,
       0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1,
       0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1])
```

In [139]:
```
import sklearn.metrics as metrics
fpr,tpr,threshold = metrics.roc_curve(y_test,y_pred)
roc_auc = metrics.auc(fpr,tpr)

import matplotlib.pyplot as plt
plt.title("ROC")
plt.plot(fpr,tpr,'b',label = 'auc = %0.2f'%roc_auc)
plt.legend(loc='lower right')
plt.plot([0,1],[0,1],'r--')
plt.xlim([0,1])
plt.ylim([0,1])
plt.xlabel('fpr')
plt.ylabel('tpr')
```

Out[139]:
```
Text(0, 0.5, 'tpr')
```



In [130]:
```
from sklearn.metrics import accuracy_score
accuracy_score(y_pred,y_test)
```

Out[130]:
```
0.7987012987012987
```

In [131]:
```
from sklearn.metrics import confusion_matrix
```

In [132]:
```
cm=confusion_matrix(y_test,y_pred)
```

In [133]:
```
cm
```

Out[133]:
```
array([[88, 15],
       [16, 35]], dtype=int64)
```

In [134]:
```
#random forest tree
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier(n_estimators=100,criterion='entropy',max_depth=10,ma
```

In [135]:
```
rfc.fit(x_train,y_train)
```

```
C:\Users\PRUDHVI\anaconda3\lib\site-packages\ipykernel_launcher.py:1: DataConve
rsionWarning: A column-vector y was passed when a 1d array was expected. Please
change the shape of y to (n_samples,), for example using ravel().
  """Entry point for launching an IPython kernel.
```

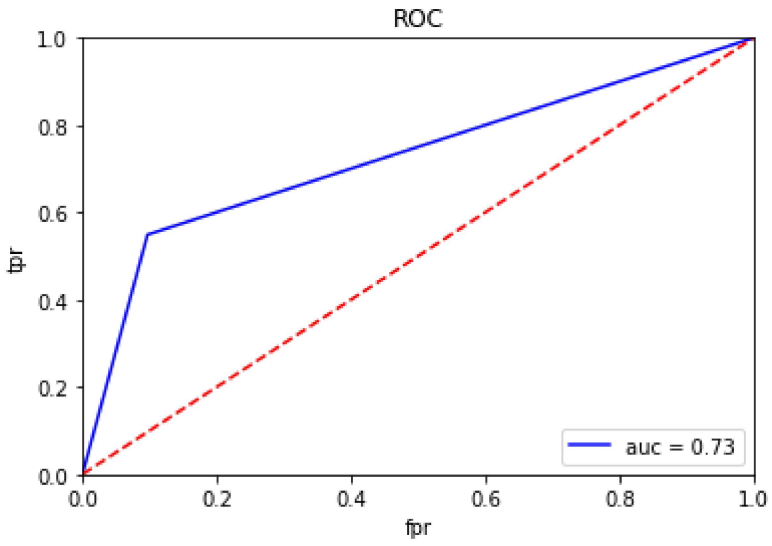
Out[135]:
```
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='entropy', max_depth=10, max_features='auto',
                       max_leaf_nodes=7, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=100,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False)
```

In [136]:
```
y_pred_rfc = rfc.predict(x_test)
```

In [140]:
```
import sklearn.metrics as metrics
fpr,tpr,threshold = metrics.roc_curve(y_test,y_pred_rfc)
roc_auc = metrics.auc(fpr,tpr)

import matplotlib.pyplot as plt
plt.title("ROC")
plt.plot(fpr,tpr,'b',label = 'auc = %0.2f'%roc_auc)
plt.legend(loc='lower right')
plt.plot([0,1],[0,1],'r--')
plt.xlim([0,1])
plt.ylim([0,1])
plt.xlabel('fpr')
plt.ylabel('tpr')
```

Out[140]:
```
Text(0, 0.5, 'tpr')
```



In [137]:
```
accuracy_score(y_test,y_pred_rfc)
```

Out[137]:
```
0.7857142857142857
```

In [141]:
```
# Logisitic regression
from sklearn.linear_model import LogisticRegression
log = LogisticRegression()
log.fit(x_train,y_train)
y_pred_log = log.predict(x_test)
y_pred_log
```

```
C:\Users\PRUDHVI\anaconda3\lib\site-packages\sklearn\utils\validation.py:760: D
ataConversionWarning: A column-vector y was passed when a 1d array was expecte
d. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

Out[141]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0,
       0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0,
       0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0,
       1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1,
       1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1,
       0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1])
```

In [142]:
```
accuracy_score(y_test,y_pred_log)
```

Out[142]:
```
0.7922077922077922
```

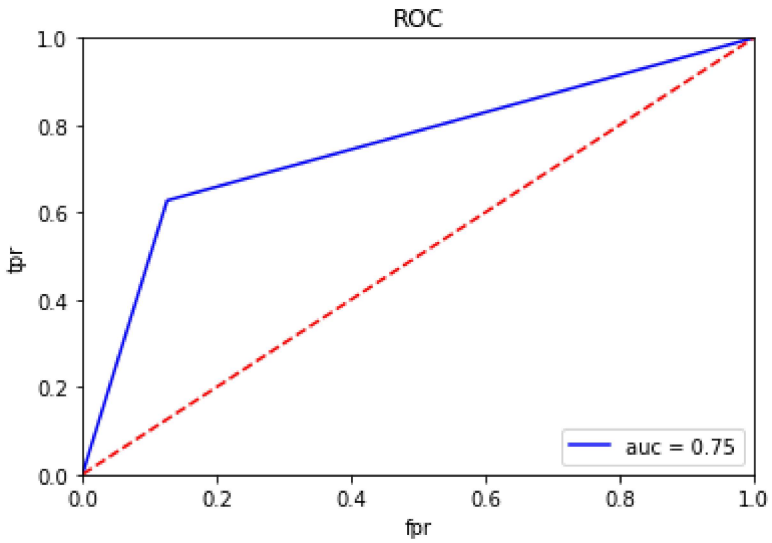
In [143]:
```
confusion_matrix(y_test,y_pred_log)
```

Out[143]:
```
array([[90, 13],
       [19, 32]], dtype=int64)
```

In [144]:
```
import sklearn.metrics as metrics
fpr,tpr,threshold = metrics.roc_curve(y_test,y_pred_log)
roc_auc = metrics.auc(fpr,tpr)

import matplotlib.pyplot as plt
plt.title("ROC")
plt.plot(fpr,tpr,'b',label = 'auc = %0.2f'%roc_auc)
plt.legend(loc='lower right')
plt.plot([0,1],[0,1],'r--')
plt.xlim([0,1])
plt.ylim([0,1])
plt.xlabel('fpr')
plt.ylabel('tpr')
```

Out[144]:
```
Text(0, 0.5, 'tpr')
```



In [153]:
```
# KNN
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors = 4)
knn.fit(x_train,y_train)
y_pred_knn = knn.predict(x_test)
y_pred_knn
```

```
C:\Users\PRUDHVI\anaconda3\lib\site-packages\ipykernel_launcher.py:4: DataConve
rsionWarning: A column-vector y was passed when a 1d array was expected. Please
change the shape of y to (n_samples, ), for example using ravel().
  after removing the cwd from sys.path.
```

Out[153]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0,
       0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0,
       1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
       0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1])
```

In [154]:
```
accuracy_score(y_test,y_pred_knn)
```

Out[154]:
```
0.7532467532467533
```

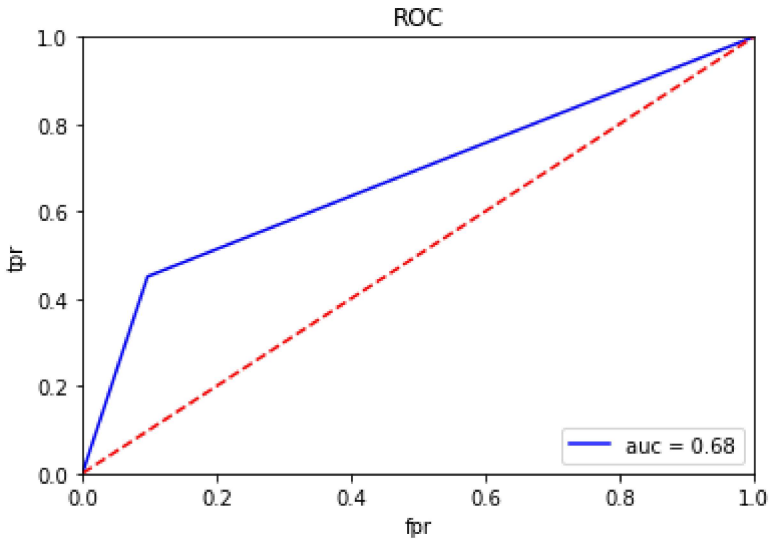
In [155]:
```
confusion_matrix(y_test,y_pred_knn)
```

Out[155]:
```
array([[93, 10],
       [28, 23]], dtype=int64)
```

In [156]:
```
import sklearn.metrics as metrics
fpr,tpr,threshold = metrics.roc_curve(y_test,y_pred_knn)
roc_auc = metrics.auc(fpr,tpr)

import matplotlib.pyplot as plt
plt.title("ROC")
plt.plot(fpr,tpr,'b',label = 'auc = %0.2f'%roc_auc)
plt.legend(loc='lower right')
plt.plot([0,1],[0,1],'r--')
plt.xlim([0,1])
plt.ylim([0,1])
plt.xlabel('fpr')
plt.ylabel('tpr')
```

Out[156]:
```
Text(0, 0.5, 'tpr')
```



In [157]:
```
# SVM
from sklearn.svm import SVC
svm = SVC(kernel = 'poly', degree = 5)
svm.fit(x_train,y_train)
y_pred_svm = svm.predict(x_test)
y_pred_svm
```

```
C:\Users\PRUDHVI\anaconda3\lib\site-packages\sklearn\utils\validation.py:760: D
ataConversionWarning: A column-vector y was passed when a 1d array was expecte
d. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

Out[157]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0,
       0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0,
       0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
       0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
       0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1])
```

In [158]:
```
accuracy_score(y_test,y_pred_svm)
```

Out[158]:
```
0.7402597402597403
```

In [159]:
```
confusion_matrix(y_test,y_pred_svm)
```

Out[159]:
```
array([[96,  7],
       [33, 18]], dtype=int64)
```

In [162]:
```
result = [['Model','Accuracy'],
          ['Decision tree',0.7987012987012987],
          ['Random Forest Classifier', 0.7857142857142857],
          ['Logistic Regression', 0.7922077922077922],
          ['K Nearest Neighbors', 0.7532467532467533],
          ['Support Vector Machine', 0.7402597402597403]]
result
```

Out[162]:
```
[['Model', 'Accuracy'],
 ['Decision tree', 0.7987012987012987],
 ['Random Forest Classifier', 0.7857142857142857],
 ['Logistic Regression', 0.7922077922077922],
 ['K Nearest Neighbors', 0.7532467532467533],
 ['Support Vector Machine', 0.7402597402597403]]
```

In [ ]:
```

```