Healthcare -Capstone Project Done by Mr.Arun Chougale

1. Data Exploration part1:

In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:
```
dataset=pd.read_csv('health care diabetes.csv')
```

In [3]:
```
dataset.head()
```

Out[3]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

In [4]:
```
dataset.shape
```

Out[4]:
```
(768, 9)
```

In [5]:
```
dataset['Outcome'].value_counts()
```

Out[5]:
```
0    500
1    268
Name: Outcome, dtype: int64
```

In [6]:
```
dataset.describe()
```

Out[6]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

Findings

Data has no negative values

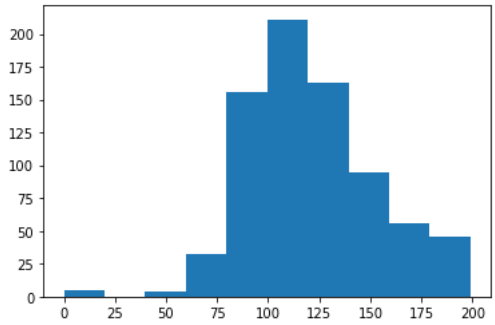
In [7]:
```
dataset.isna().sum()
```

Out[7]:
```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

We didnt get any null values here but need to check by using histograms

In [8]:
```
plt.hist(dataset['Glucose'])
```

Out[8]:
```
(array([  5.,   0.,   4.,  32., 156., 211., 163.,  95.,  56.,  46.]),
 array([  0. ,  19.9,  39.8,  59.7,  79.6,  99.5, 119.4, 139.3, 159.2,
        179.1, 199. ]),
 <BarContainer object of 10 artists>)
```



By this graph we can understood in between 19.9 to 39.8 '0' values observed and those are looking like missing values and need to treat them

Also here as per the details we can say mode value is average of 100 and 125,so we will replace this by 112.5

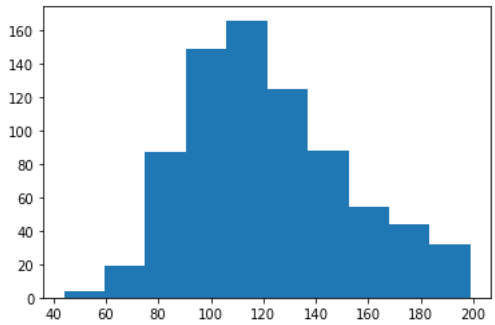
In [9]:
```
dataset['Glucose']=dataset['Glucose'].replace(0.,112.5)
```

In [10]:
```
dataset['Glucose']
```

Out[10]:
```
0      148.0
1       85.0
2      183.0
3       89.0
4      137.0
       ...
763    101.0
764    122.0
765    121.0
766    126.0
767     93.0
Name: Glucose, Length: 768, dtype: float64
```

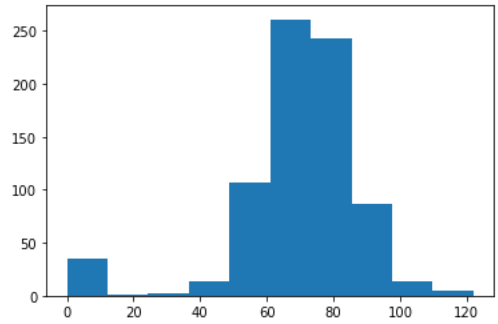
In [11]:
```
plt.hist(dataset['Glucose'])
```

Out[11]:
```
(array([  4.,  19.,  87., 149., 166., 125.,  88.,  54.,  44.,  32.]),
 array([ 44. ,  59.5,  75. ,  90.5, 106. , 121.5, 137. , 152.5, 168. ,
        183.5, 199. ]),
 <BarContainer object of 10 artists>)
```



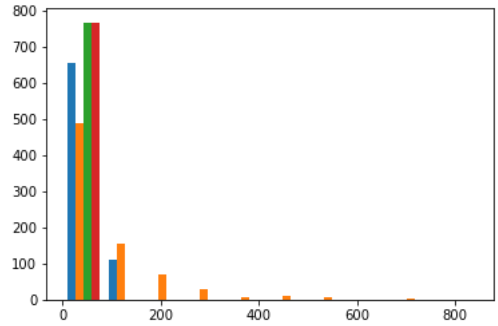
In [12]:
```
plt.hist(dataset['BloodPressure'])
```

Out[12]:
```
(array([ 35.,   1.,   2.,  13., 107., 261., 243.,  87.,  14.,   5.]),
 array([  0. ,  12.2,  24.4,  36.6,  48.8,  61. ,  73.2,  85.4,  97.6,
        109.8, 122. ]),
 <BarContainer object of 10 artists>)
```



In [13]:
```
plt.hist(dataset[['BloodPressure','Insulin','BMI','SkinThickness']])
```

Out[13]:
```
(array([[656., 112.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.],
        [487., 155.,  70.,  30.,   8.,   9.,   5.,   1.,   2.,   1.],
        [768.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.],
        [767.,   1.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.]]),
 array([  0. ,  84.6, 169.2, 253.8, 338.4, 423. , 507.6, 592.2, 676.8,
        761.4, 846. ]),
 <a list of 4 BarContainer objects>)
```



In [14]:
```
dataset['BloodPressure'].replace(0,69)
```

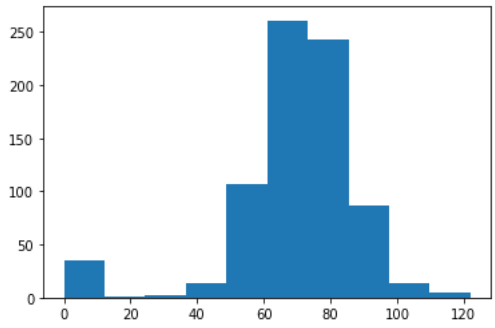
Out[14]:
```
0      72
1      66
2      64
3      66
4      40
       ..
763    76
764    70
765    72
766    60
767    70
Name: BloodPressure, Length: 768, dtype: int64
```

In [15]:
```
dataset['BloodPressure'].value_counts()
```

Out[15]:
```
70     57
74     52
78     45
68     45
72     44
64     43
80     40
76     39
60     37
0      35
62     34
66     30
82     30
88     25
84     23
90     22
86     21
58     21
50     13
56     12
52     11
54     11
75      8
92      8
65      7
85      6
94      6
48      5
96      4
44      4
100     3
106     3
98      3
110     3
55      2
108     2
104     2
46      2
30      2
122     1
95      1
102     1
61      1
24      1
38      1
40      1
114     1
Name: BloodPressure, dtype: int64
```

In [16]:
```
plt.hist(dataset['BloodPressure'])
```

Out[16]:
```
(array([ 35.,   1.,   2.,  13., 107., 261., 243.,  87.,  14.,   5.]),
 array([  0. ,  12.2,  24.4,  36.6,  48.8,  61. ,  73.2,  85.4,  97.6,
        109.8, 122. ]),
 <BarContainer object of 10 artists>)
```



In [17]:
```
dataset['Glucose'].value_counts().head(20)
```

Out[17]:
```
99.0     17
100.0    17
111.0    14
129.0    14
125.0    14
106.0    14
112.0    13
108.0    13
95.0     13
105.0    13
102.0    13
122.0    12
109.0    12
117.0    11
124.0    11
90.0     11
107.0    11
128.0    11
120.0    11
119.0    11
Name: Glucose, dtype: int64
```

In [18]:
```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    float64
 2   BloodPressure             768 non-null    int64
 3   SkinThickness             768 non-null    int64
 4   Insulin                   768 non-null    int64
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64
 8   Outcome                   768 non-null    int64
dtypes: float64(3), int64(6)
memory usage: 54.1 KB
```

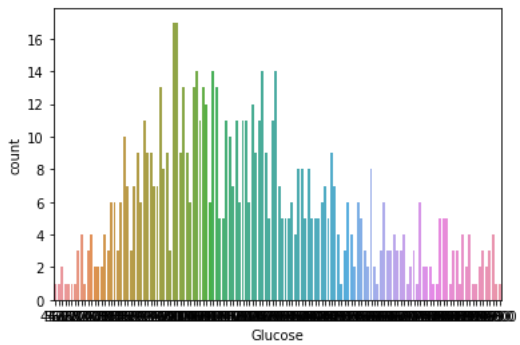
In [19]:
```
dataset.dtypes
```

Out[19]:
```
Pregnancies                 int64
Glucose                   float64
BloodPressure               int64
SkinThickness               int64
Insulin                     int64
BMI                       float64
DiabetesPedigreeFunction  float64
Age                         int64
Outcome                     int64
dtype: object
```

In [20]:
```
sns.countplot('Glucose',data=dataset)
```

```
C:\Users\Admin\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword a
rg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit ke
yword will result in an error or misinterpretation.
  warnings.warn(
```

Out[20]:
```
<AxesSubplot:xlabel='Glucose', ylabel='count'>
```

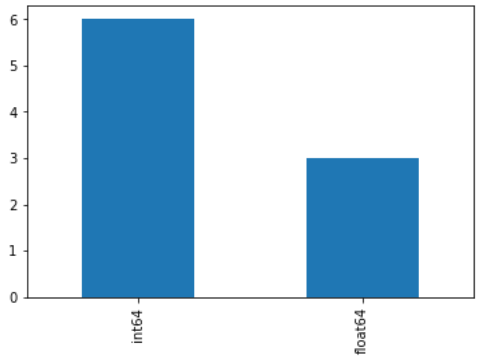


In [21]:
```
dataset['Outcome'].value_counts()
```

Out[21]:
```
0    500
1    268
Name: Outcome, dtype: int64
```

In [22]:
```
dataset.dtypes.value_counts().plot(kind='bar')
```

Out[22]: <AxesSubplot:>



Findings

1)Here I observed there only two types of attributes,one is int64 and another is float64 and how they are distributed.

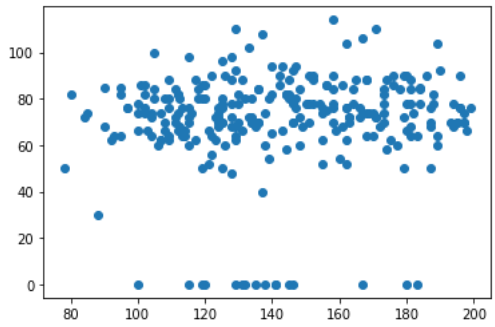
2)6 attributes are int64 and 3 attributes are float64.

Data Exploration part-2

In [23]:
```
data1=dataset[dataset['Outcome']==1]
data0=dataset[dataset['Outcome']==0]
```

In [24]:
```
plt.scatter(data1['Glucose'],data1['BloodPressure'])
```

Out[24]: <matplotlib.collections.PathCollection at 0x29434182250>

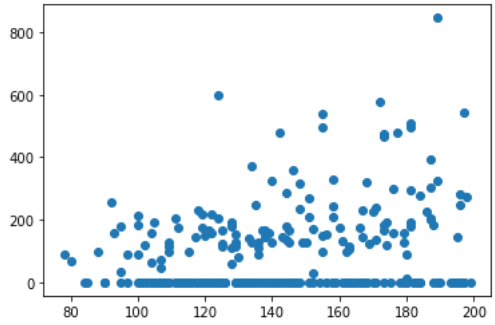


Findings-

The person who has blood-pressure above 60 ,may have diabeties too.

In [25]:
```
plt.scatter(data1['Glucose'],data1['Insulin'])
```

Out[25]:
```
<matplotlib.collections.PathCollection at 0x294341e3610>
```

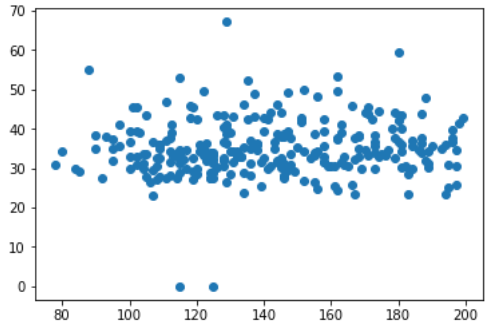


Findings-

The person who have insulin lower side ,they may have diabeties.

In [26]:
```
plt.scatter(data1['Glucose'],data1['BMI'])
```

Out[26]:
```
<matplotlib.collections.PathCollection at 0x29434250880>
```

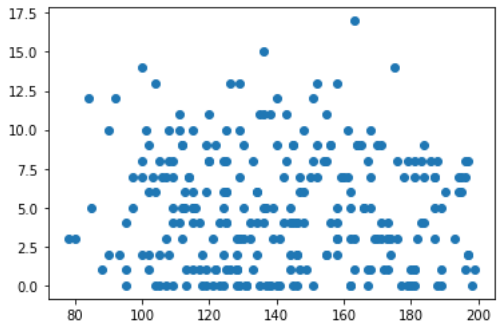


Findings-

The person whose BMI is above 30,may be have some diabetic condition with him/her.

In [27]:
```
plt.scatter(data1['Glucose'],data1['Pregnancies'])
```

Out[27]:
```
<matplotlib.collections.PathCollection at 0x29435287cd0>
```



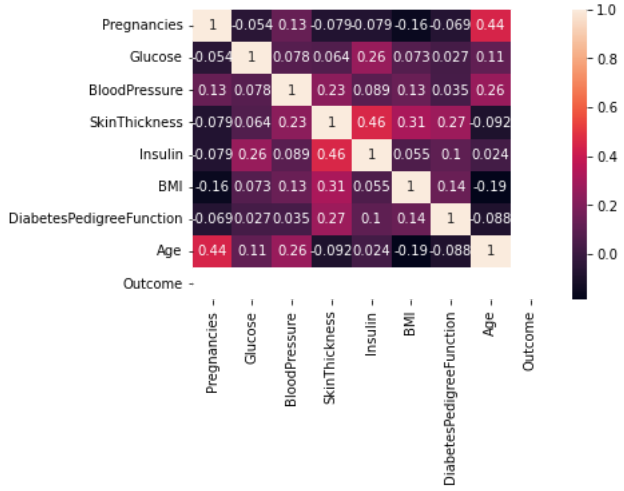
Findings-

Normally in the pregnancies,glocose level increases.

But there is no relation between number of pregnancies and diabeties.

In [28]:
```
sns.heatmap(data1.corr(),annot=True)
```

Out[28]: <AxesSubplot:>



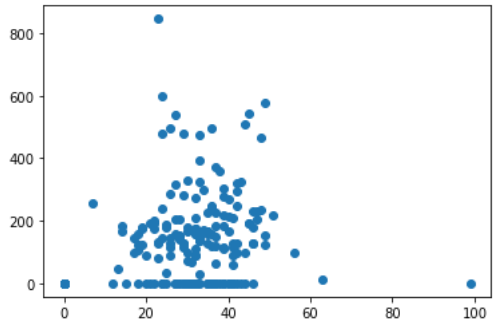
###Findings-

Skinthikness and Insuline have more correlation.

so, need to observe its realtion by scatter plot

In [29]:
```
plt.scatter(data1['SkinThickness'],data1['Insulin'])
```

Out[29]: <matplotlib.collections.PathCollection at 0x29435447ac0>



Finding-

Theperson whose skinthckness between 20 and 40 ,have less insuline in his body.

In [30]:
```
data1.shape
```

Out[30]: (268, 9)

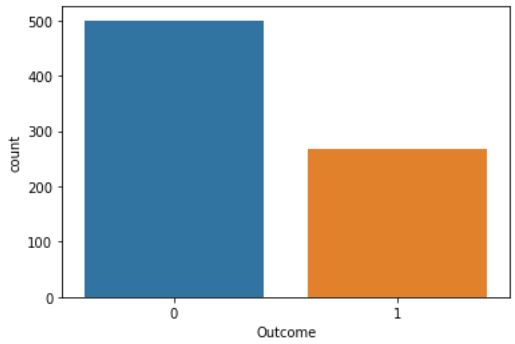
In [31]:
```
data0.shape
```

Out[31]: (500, 9)

In [32]:
```
sns.countplot(dataset['Outcome'])
```

```
C:\Users\Admin\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword a
rg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit ke
yword will result in an error or misinterpretation.
  warnings.warn(
```

Out[32]: <AxesSubplot:xlabel='Outcome', ylabel='count'>



Findings-

dataset is imbalanced

Need to do data augmentation or data resampling for balanced data

In [34]:
```
#!pip install imblearn
```

In [35]:
```
from imblearn.under_sampling import RandomUnderSampler
from imblearn.over_sampling import RandomOverSampler, SMOTE
```

In [36]:
```
print("The Number of Samples in the dataset: ", len(dataset))
print('Class 0        :', round(dataset['Outcome'].value_counts()[0]
                      /len(dataset) * 100, 2), '% of the dataset')

print('Class 1(Fraud) :', round(dataset['Outcome'].value_counts()[1]
                      /len(dataset) * 100, 2), '% of the dataset')
```

```
The Number of Samples in the dataset:  768
Class 0        : 65.1 % of the dataset
Class 1(Fraud) : 34.9 % of the dataset
```

In [37]:
```
X_data=dataset.iloc[:,:-1]
Y_data=dataset.iloc[:,-1:]
```

In [38]:
```
dataset['Outcome'].value_counts()
```

Out[38]:
```
0    500
1    268
Name: Outcome, dtype: int64
```

In [42]:
```
rus=RandomUnderSampler(random_state=42)
```

In [43]:
```
X_res,Y_res=rus.fit_resample(X_data,Y_data)

X_res = pd.DataFrame(X_res)
Y_res = pd.DataFrame(Y_res)

print("After Under Sampling Of Major Class Total Samples are :", len(Y_res))
print('Class 0:', round(Y_res.value_counts()[0]
                /len(Y_res) * 100, 2), '% of the dataset')

print('Class 1(Fraud):', round(Y_res.value_counts()[1]
                /len(Y_res) * 100, 2), '% of the dataset')
```

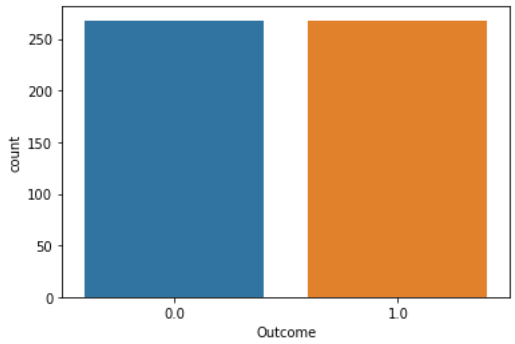
```
After Under Sampling Of Major Class Total Samples are : 536
Class 0: 50.0 % of the dataset
Class 1(Fraud): 50.0 % of the dataset
```

In [44]:
```
resampled_data=pd.concat([X_res,Y_res])
```

In [45]:
```
sns.countplot(resampled_data['Outcome'])
```

```
C:\Users\Admin\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword a
rg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit ke
yword will result in an error or misinterpretation.
  warnings.warn(
```

Out[45]: <AxesSubplot:xlabel='Outcome', ylabel='count'>



1.Now here we can data set is balanced,i.e,50%of class 0 and 50%of class 1

2.For getting balanced data we use here resampling technique of undersampling,we can use oversampling too.

3.For building a model now we are going to use balanced dataset.

In [46]:
```
Y_res.shape
```

Out[46]: (536, 1)

In [47]:
```
X_res
```

Out[47]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
0	1	97.0	70	40	0	38.1	0.218	30
1	5	78.0	48	0	0	33.7	0.654	25
2	3	111.0	58	31	44	29.5	0.430	22
3	2	129.0	84	0	0	28.0	0.284	27
4	7	102.0	74	40	105	37.2	0.204	45
...	...	...	...	...	...	...	...	...
531	1	128.0	88	39	110	36.5	1.057	37
532	0	123.0	72	0	0	36.3	0.258	52
533	6	190.0	92	0	0	35.5	0.278	66
534	9	170.0	74	31	0	44.0	0.403	43
535	1	126.0	60	0	0	30.1	0.349	47

536 rows × 8 columns

Data Modelling-part1

In [48]:
```
from sklearn.model_selection import train_test_split
```

In [49]:
```
x_train,x_test,y_train,y_test=train_test_split(X_res,Y_res,test_size=0.3,random_state=42)
```

In [50]:
```
x_train
```

Out[50]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
57	3	180.0	64	25	70	34.0	0.271	26
227	6	165.0	68	26	168	33.6	0.631	49
24	6	111.0	64	39	0	34.2	0.260	24
17	0	100.0	88	60	110	46.8	0.962	31
210	2	123.0	48	32	165	42.1	0.520	26
...	...	...	...	...	...	...	...	...
71	0	111.0	65	0	0	24.6	0.660	31
106	8	126.0	74	38	75	25.9	0.162	39
270	0	137.0	40	35	168	43.1	2.288	33
435	0	189.0	104	25	0	34.3	0.435	41
102	7	133.0	84	0	0	40.2	0.696	37

375 rows × 8 columns

In [51]:
```
x_test
```

Out[51]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
117	1	109.0	60	8	182	25.4	0.947	21
132	4	83.0	86	19	0	29.3	0.317	34
154	6	80.0	80	36	0	39.8	0.177	28
245	4	151.0	90	38	0	29.7	0.294	36
84	3	96.0	78	39	0	37.3	0.238	40
...	...	...	...	...	...	...	...	...
31	3	125.0	58	0	0	31.6	0.151	24
113	1	103.0	80	11	82	19.4	0.491	22
496	6	115.0	60	39	0	33.7	0.245	40
371	10	101.0	86	37	0	45.6	1.136	38
483	0	119.0	0	0	0	32.4	0.141	24

161 rows × 8 columns

In [52]:
```
from sklearn.tree import DecisionTreeClassifier
```

In [53]:
```
CDT=DecisionTreeClassifier(criterion='gini')
```

In [54]:
```
CDT.fit(x_train,y_train)
```

Out[54]:
```
DecisionTreeClassifier()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [55]:
```
y_pred=CDT.predict(x_test)
```

In [56]:
```
y_pred
```

Out[56]:
```
array([0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1,
       0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1,
       0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1,
       0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0,
       0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1,
       0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1,
       1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0,
       0, 0, 1, 0, 0, 1, 0], dtype=int64)
```

PERFORMANCE ANALYSIS

In [57]:
```
from sklearn.metrics import confusion_matrix,classification_report,accuracy_score
```

In [58]:
```
print("Confusion_matrix",classification_report(y_pred,y_test))
```

```
Confusion_matrix               precision    recall  f1-score   support

           0       0.82      0.67      0.74        92
           1       0.65      0.80      0.71        69

    accuracy                           0.73       161
   macro avg       0.73      0.74      0.73       161
weighted avg       0.74      0.73      0.73       161
```

In [59]:
```
print(accuracy_score(y_pred,y_test))
```

```
0.7267080745341615
```

In [60]:
```
x_pred=CDT.predict(x_train)
```

In [61]:
```
print(accuracy_score(x_pred,y_train))
```

```
1.0
```

Now I observing here overfitting condition because,this model has accuracy for train data is1 but for test data is 0.7, so to avoid this overfitting condition we have to apply below techninques

a.Pruning the DT

b.Entrpy instead of Gini

c.Crossvalidation

d.GridsearchCV

b.Entropy instead of gini

In [62]:
```
CDT=DecisionTreeClassifier(criterion='entropy')
```

In [63]:
```
CDT.fit(x_train,y_train)
```

Out[63]:
```
DecisionTreeClassifier(criterion='entropy')
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [64]:
```
y_pred_entropy=CDT.predict(x_test)
```

In [65]:
```
print(accuracy_score(y_pred_entropy,y_test))
```

```
0.7329192546583851
```

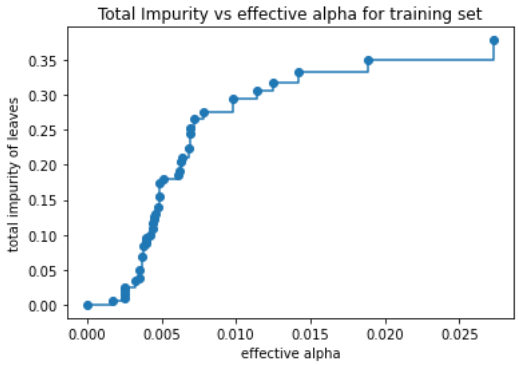
so,here I observed by using criterion entropy instead of gini,some improvement in accuracy but not as expected

a.Pruining

In [66]:
```
path=DecisionTreeClassifier(random_state=1).\
cost_complexity_pruning_path(x_train, y_train)
ccp_alphas, impurities = path.ccp_alphas, path.impurities
```

In [67]:
```
fig, ax = plt.subplots()
ax.plot(ccp_alphas[:-1], impurities[:-1], marker='o', drawstyle="steps-post")
ax.set_xlabel("effective alpha")
ax.set_ylabel("total impurity of leaves")
ax.set_title("Total Impurity vs effective alpha for training set")
```

Out[67]:
```
Text(0.5, 1.0, 'Total Impurity vs effective alpha for training set')
```



As it is evident from the above plot, the zero value of alpha corresponds to minimum impurity(unpruned tree) and as the value of alpha tends to infinity the tree tends to be more impure. ¶

Next, we build a forest of trees with different values of ccp_alpha values extracted from cost_complexity_pruning_path in order. The last tree will be the root node.

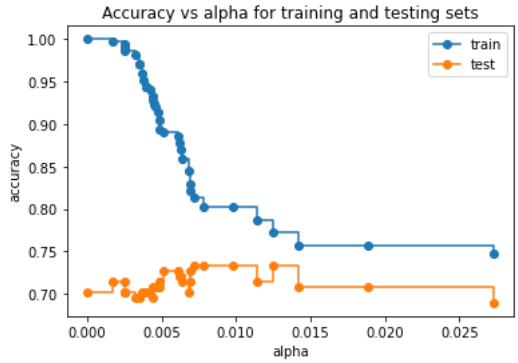
In [68]:
```
clfs = []
for ccp_alpha in ccp_alphas:
    clf = DecisionTreeClassifier(random_state=1,ccp_alpha=ccp_alpha)
    clf.fit(x_train, y_train)
    clfs.append(clf)
```

In [69]:
```
print("Number of nodes in the last tree is: {} with ccp_alpha: {} and a depth of: {}".format(
clfs[-1].tree_.node_count, ccp_alphas[-1],clfs[-1].tree_.max_depth))
```

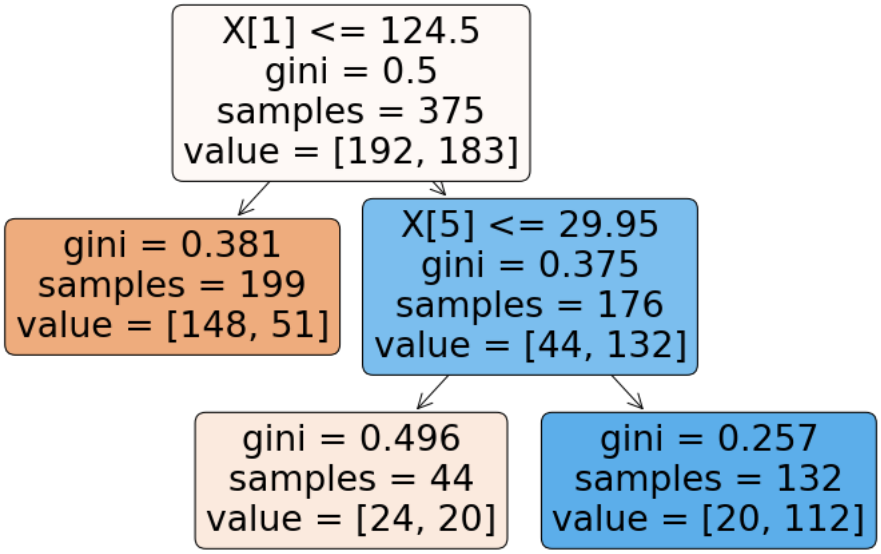
```
Number of nodes in the last tree is: 1 with ccp_alpha: 0.121420542713568 and a depth of: 0
```

In [70]:
```
clfs = clfs[:-1]
ccp_alphas = ccp_alphas[:-1]
```

In [71]:
```
train_scores = [clf.score(x_train, y_train) for clf in clfs]
test_scores = [clf.score(x_test, y_test) for clf in clfs]
fig, ax = plt.subplots()
ax.set_xlabel("alpha")
ax.set_ylabel("accuracy")
ax.set_title("Accuracy vs alpha for training and testing sets")
ax.plot(ccp_alphas, train_scores, marker='o', label="train",drawstyle="steps-post")
ax.plot(ccp_alphas, test_scores, marker='o', label="test",drawstyle="steps-post")
ax.legend()
plt.show()
```



In [72]:
```
import sklearn.tree as tree
clf=DecisionTreeClassifier(random_state=0,ccp_alpha=0.02)
clf.fit(x_train,y_train)
plt.figure(figsize=(12,8))
tree.plot_tree(clf,rounded=True,filled=True)
plt.show()
```



In [73]:
```
accuracy_score(y_test,clf.predict(x_test))
```

Out[73]:
```
0.7080745341614907
```

c)CROSS VALIDATION

In [74]:
```
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import LogisticRegression
#by kfold method
# apply classifier
clf = LogisticRegression()
# get cv scores
cv_scores = cross_val_score(clf,x_train,y_train, cv = 5)
print('Cross validation scores (5 folds): {}'.format(cv_scores))
print('The average cross validation score (5 folds): {}'.format(np.mean(cv_scores)))
## final result ##
```

```
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\utils\validation.py:1111: DataConversionWarning: A column-vector y was passe
d when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:444: ConvergenceWarning: lbfgs failed to converge
(status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html (https://scikit-learn.org/stable/modules/preprocessing.html)
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression (https://scikit-learn.org/stable/modules/line
ar_model.html#logistic-regression)
  n_iter_i = _check_optimize_result(
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\utils\validation.py:1111: DataConversionWarning: A column-vector y was passe
d when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:444: ConvergenceWarning: lbfgs failed to converge
(status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html (https://scikit-learn.org/stable/modules/preprocessing.html)
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression (https://scikit-learn.org/stable/modules/line
ar_model.html#logistic-regression)
  n_iter_i = _check_optimize_result(
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\utils\validation.py:1111: DataConversionWarning: A column-vector y was passe
d when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:444: ConvergenceWarning: lbfgs failed to converge
(status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html (https://scikit-learn.org/stable/modules/preprocessing.html)
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression (https://scikit-learn.org/stable/modules/line
ar_model.html#logistic-regression)
  n_iter_i = _check_optimize_result(
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\utils\validation.py:1111: DataConversionWarning: A column-vector y was passe
d when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:444: ConvergenceWarning: lbfgs failed to converge
(status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html (https://scikit-learn.org/stable/modules/preprocessing.html)
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression (https://scikit-learn.org/stable/modules/line
ar_model.html#logistic-regression)
  n_iter_i = _check_optimize_result(
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\utils\validation.py:1111: DataConversionWarning: A column-vector y was passe
d when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)

Cross validation scores (5 folds): [0.69333333 0.76       0.84       0.74666667 0.81333333]
The average cross validation score (5 folds): 0.7706666666666667

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:444: ConvergenceWarning: lbfgs failed to converge
(status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html (https://scikit-learn.org/stable/modules/preprocessing.html)
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression (https://scikit-learn.org/stable/modules/line
ar_model.html#logistic-regression)
  n_iter_i = _check_optimize_result(
```

D) GridSearchCv

In [75]:
```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.svm import SVC
```

In [76]:
```
param_grid = {'max_features': ['auto', 'sqrt', 'log2'],
              'ccp_alpha': [0.1, .01, .001],
              'max_depth' : [5, 6, 7, 8, 9],
              'criterion' :['gini', 'entropy']
             }
gscv_DT=GridSearchCV(estimator=CDT,param_grid=param_grid,cv=5,verbose=0)
gscv_DT.fit(x_train,y_train)
```

```
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\tree\_classes.py:298: FutureWarning: `max_features='auto'` has been deprec
ated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\tree\_classes.py:298: FutureWarning: `max_features='auto'` has been deprec
ated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\tree\_classes.py:298: FutureWarning: `max_features='auto'` has been deprec
ated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\tree\_classes.py:298: FutureWarning: `max_features='auto'` has been deprec
ated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\tree\_classes.py:298: FutureWarning: `max_features='auto'` has been deprec
ated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\tree\_classes.py:298: FutureWarning: `max_features='auto'` has been deprec
ated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\tree\_classes.py:298: FutureWarning: `max_features='auto'` has been deprec
```

In [77]:
```
gscv_DT.best_params_
```

Out[77]:
```
{'ccp_alpha': 0.01,
 'criterion': 'gini',
 'max_depth': 8,
 'max_features': 'sqrt'}
```

In [78]:
```
gscv_DT.best_score_
```

Out[78]:
```
0.7413333333333332
```

In [79]:
```
GSCV_DT=DecisionTreeClassifier(ccp_alpha=0.01,
 criterion='gini',
 max_depth=8,
 max_features='auto')
```

In [80]:
```
GSCV_DT.fit(x_train,y_train)
```

```
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\tree\_classes.py:298: FutureWarning: `max_features='auto'` has been deprecat
ed in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
```

Out[80]:
```
DecisionTreeClassifier(ccp_alpha=0.01, max_depth=8, max_features='auto')
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [81]:
```
y_gscv_dt=GSCV_DT.predict(x_test)
```

In [82]:
```
y_gscv_dt
```

Out[82]:
```
array([0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1,
       0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1,
       0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1,
       0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0,
       1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1,
       1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0,
       1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0,
       0, 1, 1, 0, 0, 0, 0], dtype=int64)
```

In [83]:
```
print(accuracy_score(y_gscv_dt,y_test))
```

```
0.6894409937888198
```

Validating result by SVM

In [84]:
```
from sklearn.svm import SVC
```

In [85]:
```
SVM=SVC(kernel='linear', random_state=0)
```

In [86]:
```
SVM.fit(x_train,y_train)
```

```
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\utils\validation.py:1111: DataConversionWarning: A column-vector y was passe
d when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

Out[86]:
```
SVC(kernel='linear', random_state=0)
```
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [87]:
```
y_svm=SVM.predict(x_test)
```

In [88]:
```
y_svm
```

Out[88]:
```
array([0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1,
       0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0,
       0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1,
       0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0,
       0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0,
       1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0,
       1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0,
       0, 1, 0, 0, 0, 1, 0], dtype=int64)
```

In [89]:
```
print(accuracy_score(y_svm,y_test))
```

```
0.7391304347826086
```

now we will check the result after grid_search_cv

In [90]:
```
params={'c':[5,10,15],'gamma':[0.01,0.02,0.05]}
gscv_svm_2=GridSearchCV(estimator=SVM,param_grid=params,cv=5,verbose=0)
gscv_svm_2.fit(x_train,y_train)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Input In [90], in <cell line: 3>()
      1 params={'c':[5,10,15],'gamma':[0.01,0.02,0.05]}
      2 gscv_svm_2=GridSearchCV(estimator=SVM,param_grid=params,cv=5,verbose=0)
----> 3 gscv_svm_2.fit(x_train,y_train)

File ~\anaconda3\lib\site-packages\sklearn\model_selection\_search.py:875, in BaseSearchCV.fit(self, X, y, groups, **fit_para
ms)
    869     results = self._format_results(
    870         all_candidate_params, n_splits, all_out, all_more_results
    871     )
    873     return results
--> 875 self._run_search(evaluate_candidates)
    877 # multimetric is determined here because in the case of a callable
    878 # self.scoring the return type is only known after calling
    879 first_test_score = all_out[0]["test_scores"]

File ~\anaconda3\lib\site-packages\sklearn\model_selection\_search.py:1379, in GridSearchCV._run_search(self, evaluate_candid
```

In [ ]:
```
#Now we have to follow the same steps with different algorithms
#KNN,RF,XGB,
```

In [ ]:
```
## save the moodels with joblib,pickle
```

In [92]:
```
from sklearn.neighbors import KNeighborsClassifier
knn_classifier= KNeighborsClassifier(n_neighbors=5, metric='minkowski', p=2 )
knn_classifier.fit(x_train, y_train)
```

```
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:207: DataConversionWarning: A column-vector y w
as passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
  return self._fit(X, y)
```

Out[92]:
```
KNeighborsClassifier()
```
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [93]:
```
y_pred_knn_classifier=knn_classifier.predict(x_test)
```

In [94]:
```
print(accuracy_score(y_pred_knn_classifier,y_test))
```

```
0.7267080745341615
```

Now we check with Grid_searchCV

In [95]:
```
k_range = list(range(1, 31))
param_grid = dict(n_neighbors=k_range)
grid = GridSearchCV(knn_classifier, param_grid, cv=10, scoring='accuracy', return_train_score=False,verbose=1)
```

In [96]:
```
grid_search=grid.fit(x_train,y_train)
```

```
Fitting 10 folds for each of 30 candidates, totalling 300 fits

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:207: DataConversionWarning: 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().
  return self._fit(X, y)
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:207: DataConversionWarning: 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().
  return self._fit(X, y)
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:207: DataConversionWarning: 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().
  return self._fit(X, y)
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:207: DataConversionWarning: 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().
  return self._fit(X, y)
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:207: DataConversionWarning: 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().
  return self._fit(X, y)
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:207: DataConversionWarning: 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().
  return self._fit(X, y)
```

In [97]:
```
grid_search.best_params_
```

Out[97]:
```
{'n_neighbors': 9}
```

In [98]:
```
knn_classifier= KNeighborsClassifier(n_neighbors=9 )
```

In [99]:
```
knn_classifier.fit(x_train,y_train)
```

```
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:207: DataConversionWarning: A column-vector y w
as passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
  return self._fit(X, y)
```

Out[99]:
```
KNeighborsClassifier(n_neighbors=9)
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [100]:
```
y_gscv_knn=knn_classifier.predict(x_test)
```

In [101]:
```
print(accuracy_score(y_gscv_knn,y_test))
```

```
0.7515527950310559
```

In [102]:
```
from sklearn.ensemble import RandomForestClassifier
```

In [103]:
```
rf=RandomForestClassifier(random_state=42)
```

In [104]:
```
rf.fit(x_train,y_train)
```

```
C:\Users\Admin\AppData\Local\Temp\ipykernel_1892\1149647727.py:1: DataConversionWarning: 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().
  rf.fit(x_train,y_train)
```

Out[104]:
```
RandomForestClassifier(random_state=42)
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [105]:
```
y_rf=rf.predict(x_test)
```

In [106]:
```
print('Accuracy score of Random Forest without GridDSearchCV:',accuracy_score(y_rf,y_test))
```

```
Accuracy score of Random Forest without GridDSearchCV: 0.7950310559006211
```

In [107]:
```
# Now with the help of gridsearchCV we are going to find best fit parameters
```

In [108]:
```
param_grid = {
    'n_estimators': [200, 500],
    'max_features': ['auto', 'sqrt', 'log2'],
    'max_depth' : [4,5,6,7,8],
    'criterion' :['gini', 'entropy']
}
CV_rfc = GridSearchCV(estimator=rf, param_grid=param_grid, cv= 5)
CV_rfc.fit(x_train,y_train)
```

```
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:686: DataConversionWarning: 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().
  estimator.fit(X_train, y_train, **fit_params)
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:427: FutureWarning: `max_features='auto'` has been dep
recated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features='sqrt'` or remove this pa
rameter as it is also the default value for RandomForestClassifiers and ExtraTreesClassifiers.
  warn(
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:686: DataConversionWarning: 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().
  estimator.fit(X_train, y_train, **fit_params)
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:427: FutureWarning: `max_features='auto'` has been dep
recated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features='sqrt'` or remove this pa
rameter as it is also the default value for RandomForestClassifiers and ExtraTreesClassifiers.
  warn(
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:686: DataConversionWarning: 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().
  estimator.fit(X_train, y_train, **fit_params)
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:427: FutureWarning: `max_features='auto'` has been dep
recated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features='sqrt'` or remove this pa
```

In [ ]:
```
CV_rfc.best_params_
```

In [ ]:
```
rf_CV=RandomForestClassifier( n_estimators=500,
    criterion='gini',
    max_depth=4,
    min_samples_split=2,
    min_samples_leaf=1,
    min_weight_fraction_leaf=0.0,
    max_features='auto',
    max_leaf_nodes=None,
    min_impurity_decrease=0.0,
    bootstrap=True,
    oob_score=False,
    n_jobs=None,
    random_state=42,
    verbose=0,
    warm_start=False,
    class_weight=None,
    ccp_alpha=0.0,
    max_samples=None)
```

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

In [ ]:
```
y_pred_rfcv=rf_CV.predict(x_test)
```

In [ ]:
```
print('Accuracy score of Random Forest with GridDSearchCV:',accuracy_score(y_pred_rfcv,y_test))
```

XGB boosting algorithm

In [109]:
```
import xgboost as xgb
```

In [110]:
```
XGB = xgb.XGBClassifier()
XGB.fit(x_train, y_train)
```

Out[110]:
```
XGBClassifier(base_score=0.5, booster='gbtree', callbacks=None,
              colsample_bylevel=1, colsample_bynode=1, colsample_bytree=1,
              early_stopping_rounds=None, enable_categorical=False,
              eval_metric=None, gamma=0, gpu_id=-1, grow_policy='depthwise',
              importance_type=None, interaction_constraints='',
              learning_rate=0.300000012, max_bin=256, max_cat_to_onehot=4,
              max_delta_step=0, max_depth=6, max_leaves=0, min_child_weight=1,
              missing=nan, monotone_constraints='()', n_estimators=100,
              n_jobs=0, num_parallel_tree=1, predictor='auto', random_state=0,
              reg_alpha=0, reg_lambda=1, ...)
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [111]:
```
y_xgb=XGB.predict(x_test)
```

In [112]:
```
y_xgb
```

Out[112]:
```
array([0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1,
       0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1,
       0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1,
       0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0,
       0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0,
       0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1,
       1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0,
       0, 1, 1, 0, 1, 1, 1])
```

In [113]:
```
print('Accuracy_score of XGB Algorithm:',accuracy_score(y_xgb,y_test))
```

```
Accuracy_score of XGB Algorithm: 0.7515527950310559
```

In [114]:
```
from sklearn.linear_model import LogisticRegression
```

In [115]:
```
lr=LogisticRegression()
```

In [116]:
```
lr.fit(x_train,y_train)
```

```
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\utils\validation.py:1111: DataConversionWarning: A column-vector y was passe
d when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:444: ConvergenceWarning: lbfgs failed to converge
(status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html (https://scikit-learn.org/stable/modules/preprocessing.html)
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression (https://scikit-learn.org/stable/modules/line
ar_model.html#logistic-regression)
  n_iter_i = _check_optimize_result(
```

Out[116]:
```
LogisticRegression()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [117]:
```
y_pred_lr=lr.predict(x_test)
```

In [118]:
```
print('Accuracy_score of Logistic Regression:',accuracy_score(y_pred_lr,y_test))
```

```
Accuracy_score of Logistic Regression: 0.7701863354037267
```

with GridSearchCV

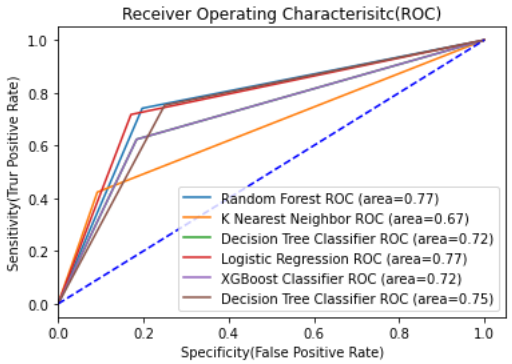
In [119]:
```
grid={"C":np.logspace(-3,3,7), "penalty":["l1","l2"]}# l1 lasso l2 ridge
logreg=LogisticRegression()
logreg_cv=GridSearchCV(logreg,grid,cv=10)
logreg_cv.fit(x_train,y_train)
```

```
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\utils\validation.py:1111: DataConversionWarning: A column-vector y was pas
sed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:444: ConvergenceWarning: lbfgs failed to converg
e (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html (https://scikit-learn.org/stable/modules/preprocessing.html)
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression (https://scikit-learn.org/stable/modules/li
near_model.html#logistic-regression)
  n_iter_i = _check_optimize_result(
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\utils\validation.py:1111: DataConversionWarning: A column-vector y was pas
sed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:444: ConvergenceWarning: lbfgs failed to converg
e (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.
```

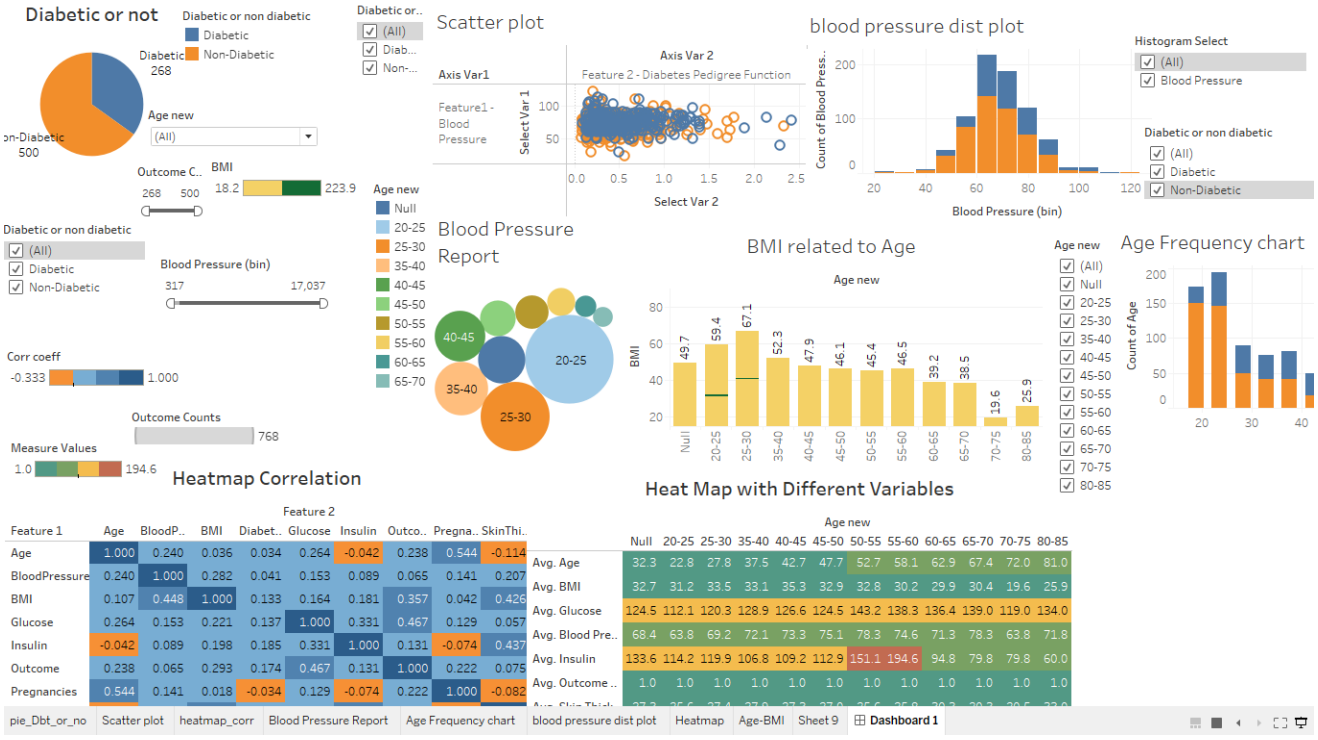
AUC curves for various Algorithms

In [127]:
```
import sklearn.metrics as metrics
plt.figure()
models=[
    { 'label':'Random Forest',
      'model':RandomForestClassifier()
    },
    {
        'label':'K Nearest Neighbor',
        'model':KNeighborsClassifier(n_neighbors=2)
    },
    {
        'label':'Decision Tree Classifier',
        'model':DecisionTreeClassifier()
    },
    {
        'label':'Logistic Regression',
        'model':LogisticRegression()
    },

    {
        'label':'XGBoost Classifier',
        'model':DecisionTreeClassifier()
    },
    {
        'label':'Decision Tree Classifier',
        'model':xgb.XGBClassifier()
    }

]

for mod in models:
    model=mod['model']
    model.fit(x_train,y_train)
    y_pred=model.predict(x_test)


    fpr,tpr,thresholds=metrics.roc_curve(y_test,y_pred)

    auc=metrics.roc_auc_score(y_test,model.predict(x_test))

    plt.plot(fpr,tpr,label='%s ROC (area=%0.2f)'%(mod['label'],auc))



plt.plot([0,1],[0,1],'b--')
plt.xlim([0.0,1.0])
plt.xlim([0.0,1.05])
plt.xlabel('Specificity(False Positive Rate)')
plt.ylabel('Sensitivity(Trur Positive Rate)')
plt.title('Receiver Operating Characterisitc(ROC)')
plt.legend(loc='lower right')
plt.show()
```

```
C:\Users\Admin\AppData\Local\Temp\ipykernel_1892\1919318965.py:33: DataConversionWarning: A column-vector y was passed when a 1
d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
  model.fit(x_train,y_train)
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:207: DataConversionWarning: A column-vector y w
as passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
  return self._fit(X, y)
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\utils\validation.py:1111: DataConversionWarning: A column-vector y was passe
d when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:444: ConvergenceWarning: lbfgs failed to converge
(status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html (https://scikit-learn.org/stable/modules/preprocessing.html)
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression (https://scikit-learn.org/stable/modules/line
ar_model.html#logistic-regression)
  n_iter_i = _check_optimize_result(
```



Tableu Dashboard-Healthcare



End Of Project

Thank You.....Arun R Chougale

In [ ]: