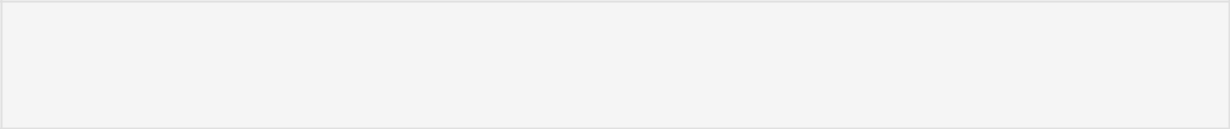
LabAssignment\_2.1 07/12/22, 11:09 PM

In [23]:

Lab Assignment No 2.1 (DecisionTreeClassifier)

Imported Pandas as SmokingDataSet



**import** pandas **as** pd

df **=** pd**.**read\_csv('SmokingDataSet.csv')

df**.**head()

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Out[23]: | | gender | age | hypertension | heart\_disease | ever\_married work\_type | | Residence\_type |  |
|  |  |  |  |  |  |  |  |  |  |
| 0 | | Male | 67.0 | 0 | 1 | Yes | Private | Urban |  |
|  | |  |  |  |  |  |  |  |  |
|  | 1 | Male | 80.0 | 0 | 1 | Yes | Private | Rural |  |
|  |  |  |  |  |  |  |  |  |  |
| 2 | | Female | 49.0 | 0 | 0 | Yes | Private | Urban |  |
|  |  |  |  |  |  |  |  |  |  |
|  | 3 | Female | 79.0 | 1 | 0 | Yes | Self- | Rural |  |
|  | employed |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| 4 | | Male | 81.0 | 0 | 0 | Yes | Private | Urban |  |

In [24]:

DataSet Information



df**.**info()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| <class 'pandas.core.frame.DataFrame'> | | | |  |
| RangeIndex: 4981 entries, 0 to 4980 | | | |  |
| Data | columns (total 11 columns): | | |  |
| # | Column | Non-Null Count | | Dtype |
| --- | ------ | -------------- | | ----- |
| 0 | gender | 4981 | non-null | object |
| 1 | age | 4981 | non-null | float64 |
| 2 | hypertension | 4981 | non-null | int64 |
| 3 | heart\_disease | 4981 | non-null | int64 |
| 4 | ever\_married | 4981 | non-null | object |
| 5 | work\_type | 4981 | non-null | object |
| 6 | Residence\_type | 4981 | non-null | object |
| 7 | avg\_glucose\_level | 4981 | non-null | float64 |
| 8 | bmi | 4981 | non-null | float64 |
| 9 | smoking\_status | 4981 | non-null | object |
| 10 | stroke | 4981 | non-null | int64 |

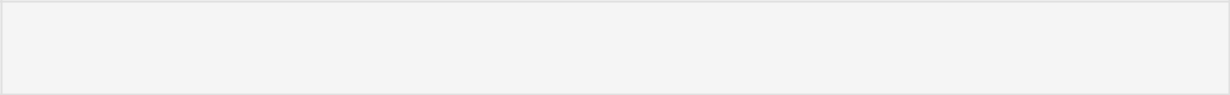
dtypes: float64(3), int64(3), object(5)

memory usage: 428.2+ KB

In [25]:

Out[25]:

select all the columns of type object and put it in list



objectList **=** list(df**.**select\_dtypes(include**=**'object'))

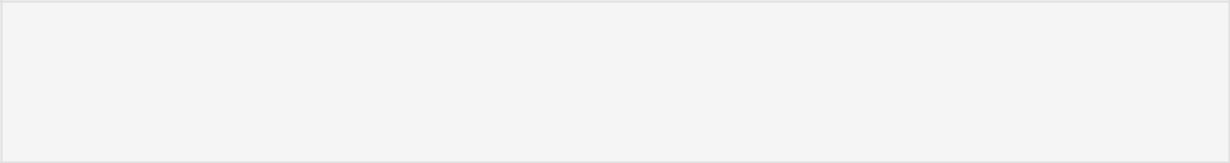
objectList

['gender', 'ever\_married', 'work\_type', 'Residence\_type', 'smoking\_status ']

Performing label Encoding on Them

http://localhost:8888/nbconvert/html/Documents/MTECH(DSA)/MyLabWork/AML\_Lab/LabAssignment\_2.1.ipynb?download=false Page 1 of 3

LabAssignment\_2.1 07/12/22, 11:09 PM



In [26]:

**from** sklearn **import** preprocessing

**for** i **in** objectList:

Encoder **=** preprocessing**.**LabelEncoder()

df[i]**=** Encoder**.**fit\_transform(df[i])

In [27]:

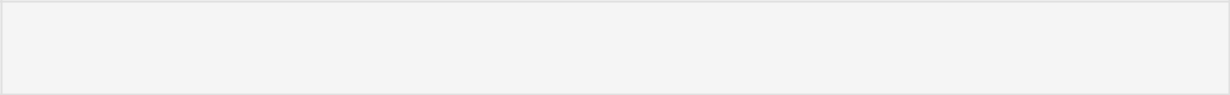
Out[27]:

Checking For Null Values Luckily no null values



df**.**isnull()**.**sum()

|  |  |
| --- | --- |
| gender | 0 |
| age | 0 |
| hypertension | 0 |
| heart\_disease | 0 |
| ever\_married | 0 |
| work\_type | 0 |
| Residence\_type | 0 |
| avg\_glucose\_level | 0 |
| bmi | 0 |
| smoking\_status | 0 |
| stroke | 0 |
| dtype: int64 |  |

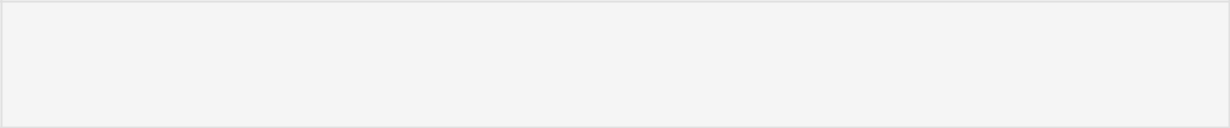


In [28]:

1. **=** df**.**drop(columns**=**['stroke'],axis**=**1)
2. **=** df['stroke']

In [29]:

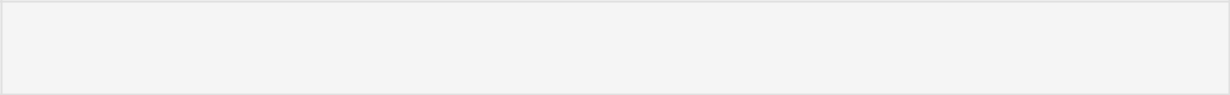
Performed OverSampling



**from** imblearn.over\_sampling **import** RandomOverSampler

over\_sampler **=** RandomOverSampler(sampling\_strategy**=**'minority')

x,y **=** over\_sampler**.**fit\_resample(x,y)

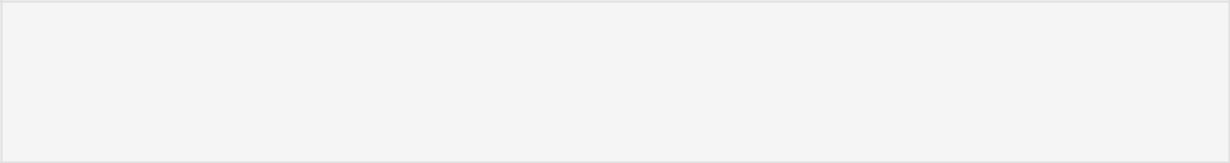


In [30]:

In [31]:

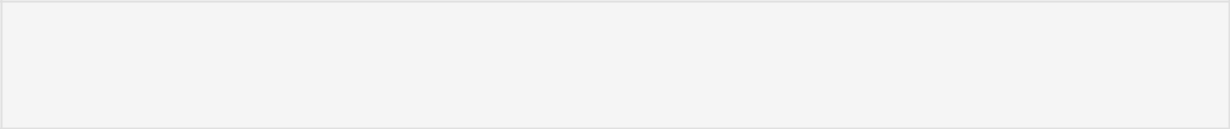
**from** sklearn.model\_selection **import** train\_test\_split

x\_train, x\_test, y\_train, y\_test **=** train\_test\_split(x, y, test\_size**=**0.20,



**from** sklearn.tree **import** DecisionTreeClassifier DT\_model **=** DecisionTreeClassifier() DT\_model**.**fit(x\_train,y\_train)

y\_pred **=** DT\_model**.**predict(x\_test)



In [32]:

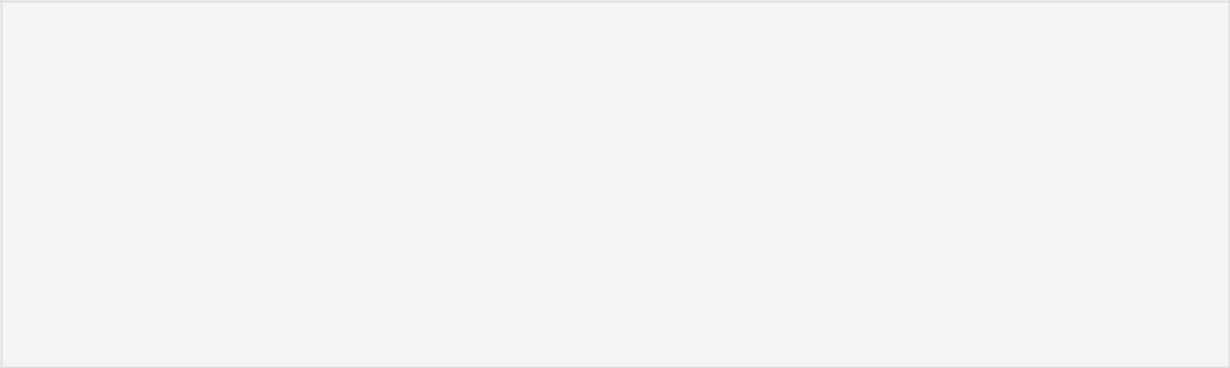
**from** sklearn.metrics **import** confusion\_matrix cm\_log **=** confusion\_matrix(y\_test,y\_pred) cm\_log

Out[32]: array([[889, 58],

[ 0, 947]])

http://localhost:8888/nbconvert/html/Documents/MTECH(DSA)/MyLabWork/AML\_Lab/LabAssignment\_2.1.ipynb?download=false Page 2 of 3

LabAssignment\_2.1 07/12/22, 11:09 PM



In [33]:

**from** sklearn.metrics **import** roc\_auc\_score, roc\_curve **import** matplotlib.pyplot **as** plt

**def** plot\_roc\_curve(y\_test,y\_pred):

fpr, tpr, thresholds **=** roc\_curve(y\_test,y\_pred)

plt**.**plot(fpr, tpr)

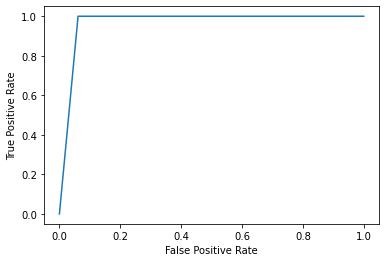
plt**.**xlabel('False Positive Rate')

plt**.**ylabel('True Positive Rate')

plot\_roc\_curve(y\_test,y\_pred)

print(f'model(DecisionTree) AUC score: {roc\_auc\_score(y\_test, y\_pred)}')

model(DecisionTree) AUC score: 0.969376979936642



http://localhost:8888/nbconvert/html/Documents/MTECH(DSA)/MyLabWork/AML\_Lab/LabAssignment\_2.1.ipynb?download=false Page 3 of 3