

```
import pandas as pd
import numpy as np
import pickle
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import sklearn
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.model_selection import RandomizedSearchCV
import imblearn
from imblearn.over_sampling import SMOTE
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix, f1_score
```

```
data = pd.read_csv("WA_Fn-UseC_-Telco-Customer-Churn.csv")
data
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	...
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	...
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	...
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	...
...	...	...	...	...	...	...	...	...	...	...	...
7038	6840-RESVB	Male	0	Yes	Yes	24	Yes	Yes	DSL	Yes	...
7039	2234-XADUH	Female	0	Yes	Yes	72	Yes	Yes	Fiber optic	No	...
7040	4801-JJAZL	Female	0	Yes	Yes	11	No	No phone service	DSL	Yes	...
7041	8361-LTMKD	Male	1	Yes	No	4	Yes	Yes	Fiber optic	No	...
7042	3186-AJIEK	Male	0	No	No	66	Yes	No	Fiber optic	Yes	...

7043 rows × 21 columns

```
data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
#   Column              Non-Null Count  Dtype
---  -
0   customerID          7043 non-null  object
1   gender              7043 non-null  object
2   SeniorCitizen       7043 non-null  int64
3   Partner             7043 non-null  object
4   Dependents          7043 non-null  object
5   tenure              7043 non-null  int64
6   PhoneService        7043 non-null  object
7   MultipleLines       7043 non-null  object
8   InternetService     7043 non-null  object
9   OnlineSecurity      7043 non-null  object
10  OnlineBackup        7043 non-null  object
11  DeviceProtection    7043 non-null  object
```

```

12 TechSupport      7043 non-null object
13 StreamingTV      7043 non-null object
14 StreamingMovies   7043 non-null object
15 Contract          7043 non-null object
16 PaperlessBilling  7043 non-null object
17 PaymentMethod     7043 non-null object
18 MonthlyCharges    7043 non-null float64
19 TotalCharges      7043 non-null object
20 Churn             7043 non-null object
dtypes: float64(1), int64(2), object(18)
memory usage: 1.1+ MB

```

```

data.TotalCharges = pd.to_numeric("data.TotalCharges", errors='coerce')
data.isnull().any()

```

```

customerID      False
gender           False
SeniorCitizen   False
Partner          False
Dependents       False
tenure           False
PhoneService     False
MultipleLines    False
InternetService  False
OnlineSecurity   False
OnlineBackup     False
DeviceProtection False
TechSupport      False
StreamingTV      False
StreamingMovies  False
Contract         False
PaperlessBilling False
PaymentMethod    False
MonthlyCharges   False
TotalCharges     True
Churn            False
dtype: bool

```

```

data.fillna(data.median)
data.isnull().sum()

```

```

customerID      0
gender           0
SeniorCitizen    0
Partner          0
Dependents       0
tenure           0
PhoneService     0
MultipleLines    0
InternetService  0
OnlineSecurity   0
OnlineBackup     0
DeviceProtection 0
TechSupport      0
StreamingTV      0
StreamingMovies  0
Contract         0
PaperlessBilling 0
PaymentMethod    0
MonthlyCharges   0
TotalCharges     7043
Churn            0
dtype: int64

```

```

from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
data["gender"] = le.fit_transform(data["gender"])
data["Partner"] = le.fit_transform(data["Partner"])
data["Dependents"] = le.fit_transform(data["Dependents"])
data["PhoneService"] = le.fit_transform(data["PhoneService"])
data["MultipleLines"] = le.fit_transform(data["MultipleLines"])
data["InternetService"] = le.fit_transform(data["InternetService"])
data["OnlineSecurity"] = le.fit_transform(data["OnlineSecurity"])
data["OnlineBackup"] = le.fit_transform(data["OnlineBackup"])
data["DeviceProtection"] = le.fit_transform(data["DeviceProtection"])
data["TechSupport"] = le.fit_transform(data["TechSupport"])
data["StreamingTV"] = le.fit_transform(data["StreamingTV"])
data["StreamingMovies"] = le.fit_transform(data["StreamingMovies"])
data["Contract"] = le.fit_transform(data["Contract"])
data["PaperlessBilling"] = le.fit_transform(data["PaperlessBilling"])

```

```
data["PaymentMethod"] = le.fit_transform(data["PaymentMethod"])
data["Churn"] = le.fit_transform(data["Churn"])
```

```
data.head()
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	Dev
0	7590-VHVEG	0	0	1	0	1	0	1	0	0	...	
1	5575-GNVDE	1	0	0	0	34	1	0	0	2	...	
2	3668-QPYBK	1	0	0	0	2	1	0	0	2	...	
3	7795-CFOCW	1	0	0	0	45	0	1	0	2	...	
4	9237-HQITU	0	0	0	0	2	1	0	1	0	...	

5 rows × 21 columns

```
x = data.iloc[:,1:19].values
y = data.iloc[:,20].values
```

x

```
array([[ 0. ,  0. ,  1. , ...,  1. ,  2. , 29.85],
       [ 1. ,  0. ,  0. , ...,  0. ,  3. , 56.95],
       [ 1. ,  0. ,  0. , ...,  1. ,  3. , 53.85],
       ...,
       [ 0. ,  0. ,  1. , ...,  1. ,  2. , 29.6 ],
       [ 1. ,  1. ,  1. , ...,  1. ,  3. , 74.4 ],
       [ 1. ,  0. ,  0. , ...,  1. ,  0. , 105.65]])
```

y

```
array([0, 0, 1, ..., 0, 1, 0])
```

```
from sklearn.preprocessing import OneHotEncoder
one = OneHotEncoder()
a= one.fit_transform(x[:,6:7]).toarray()
b= one.fit_transform(x[:,7:8]).toarray()
c= one.fit_transform(x[:,8:9]).toarray()
d= one.fit_transform(x[:,9:18]).toarray()
e= one.fit_transform(x[:,10:11]).toarray()
f= one.fit_transform(x[:,11:12]).toarray()
g= one.fit_transform(x[:,12:13]).toarray()
h= one.fit_transform(x[:,13:14]).toarray()
i= one.fit_transform(x[:,14:15]).toarray()
j= one.fit_transform(x[:,16:17]).toarray()
x=np.delete(x,[6,7,8,9,10,11,12,13,14,16], axis=1)
x=np.concatenate((a,b,c,d,e,f,g,h,i,j,x),axis=1)
```

```
from imblearn.over_sampling import SMOTE
```

```
smt = SMOTE()
x_resample, y_resample = smt.fit_resample(x,y)
```

x\_resample

```
array([[0.00000000e+00, 1.00000000e+00, 0.00000000e+00, ...,
       0.00000000e+00, 1.00000000e+00, 2.98500000e+01],
       [1.00000000e+00, 0.00000000e+00, 0.00000000e+00, ...,
       1.00000000e+00, 0.00000000e+00, 5.69500000e+01],
       [1.00000000e+00, 0.00000000e+00, 0.00000000e+00, ...,
       1.00000000e+00, 1.00000000e+00, 5.38500000e+01],
       ...,
       [1.00000000e+00, 0.00000000e+00, 0.00000000e+00, ...,
       1.00000000e+00, 1.00000000e+00, 5.08215503e+01],
       [1.24276898e-02, 9.87572310e-01, 0.00000000e+00, ...,
       1.24276898e-02, 1.00000000e+00, 4.83723698e+01],
```

```
[1.00000000e+00, 0.00000000e+00, 0.00000000e+00, ...,
1.00000000e+00, 8.12020336e-01, 4.53657929e+01]]])
```

```
y_resample
array([0, 0, 1, ..., 1, 1, 1])
```

```
x.shape, x_resample.shape
((7043, 1645), (10348, 1645))
```

```
y.shape, y_resample.shape
((7043,), (10348,))
```

```
data.describe()
```

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	On
count	7043.000000	7043.000000	7043.000000	7043.000000	7043.000000	7043.000000	7043.000000	7043.000000	7043.000000	7
mean	0.504756	0.162147	0.483033	0.299588	32.371149	0.903166	0.940508	0.872923	0.790004	
std	0.500013	0.368612	0.499748	0.458110	24.559481	0.295752	0.948554	0.737796	0.859848	
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	0.000000	0.000000	0.000000	0.000000	9.000000	1.000000	0.000000	0.000000	0.000000	
50%	1.000000	0.000000	0.000000	0.000000	29.000000	1.000000	1.000000	1.000000	1.000000	
75%	1.000000	0.000000	1.000000	1.000000	55.000000	1.000000	2.000000	1.000000	2.000000	
max	1.000000	1.000000	1.000000	1.000000	72.000000	1.000000	2.000000	2.000000	2.000000	

```
plt.figure(figsize=(12,5))
plt.subplot(1,2,1)
sns.distplot(data["tenure"])
plt.subplot(1,2,2)
sns.distplot(data["MonthlyCharges"])
```

```
<ipython-input-19-3bd718de5fe4>:3: UserWarning:
```

```
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.
```

Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

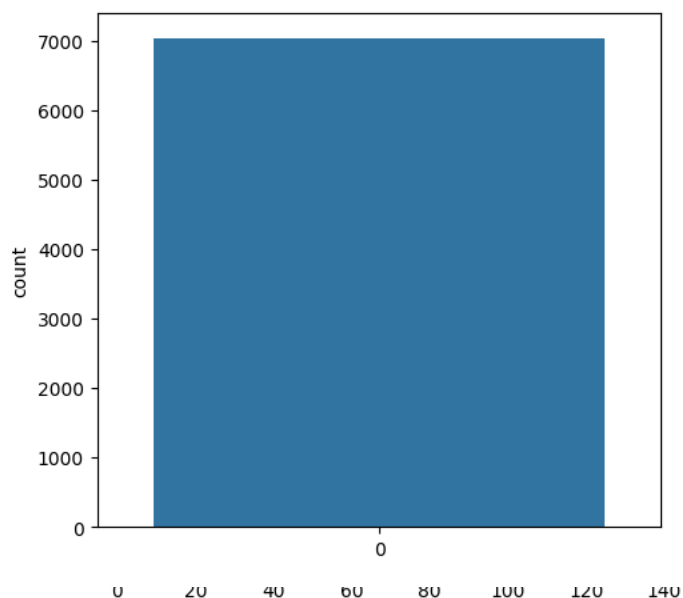
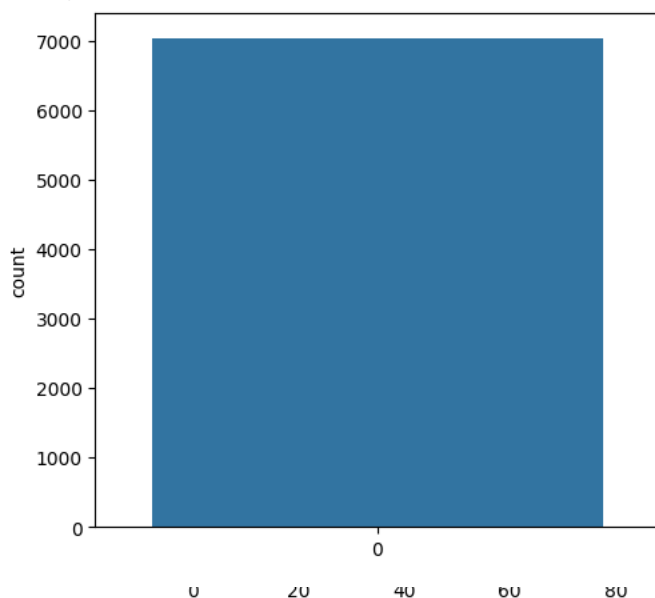
For a guide to updating your code to use the new functions, please see <https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

```
sns.distplot(data["tenure"])
```

```
<ipython-input-19-3bd718de5fe4>:5: UserWarning:
```

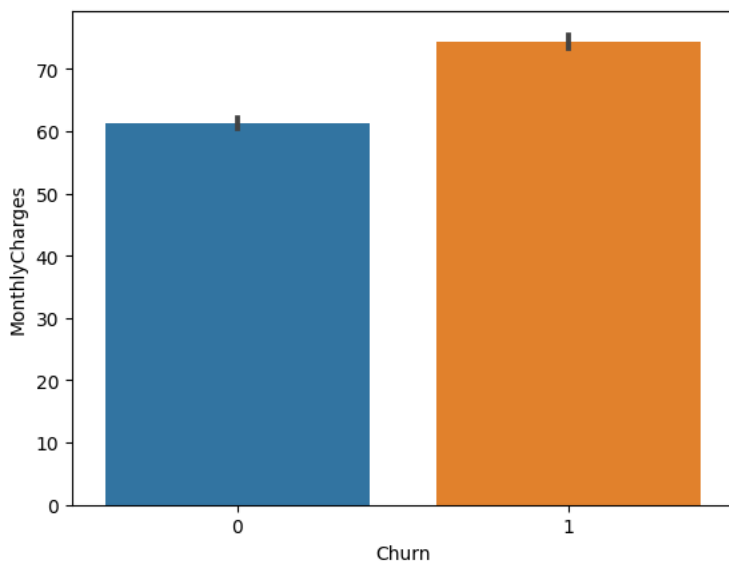
```
plt.figure(figsize=(12,5))
plt.subplot(1,2,1)
sns.countplot(data["gender"])
plt.subplot(1,2,2)
sns.countplot(data["Dependents"])
```

```
<Axes: ylabel='count'>
```



```
sns.barplot(x="Churn", y="MonthlyCharges",data=data)
```

```
<Axes: xlabel='Churn', ylabel='MonthlyCharges'>
```



```
sns.heatmap(data.corr(), annot=True)
```

[illegible]

```
sns.pairplot(data=data, markers=["^", "v"], palette="inferno")
```

[https://colab.research.google.com/drive/1jumOekuKtqbDrk0tJvTRZLoFbfRAUcAv?usp=drive\\_fs#scrollTo=oA8TXZ0mufhe&printMode=true](https://colab.research.google.com/drive/1jumOekuKtqbDrk0tJvTRZLoFbfRAUcAv?usp=drive_fs#scrollTo=oA8TXZ0mufhe&printMode=true) 7/16

```
/usr/local/lib/python3.9/dist-packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring palette because no hue variable has been ass
```

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x_resample,y_resample,test_size = 0.2, random_state = 0)

/usr/local/lib/python3.9/dist-packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring palette because no hue variable has been ass

from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_train=sc.fit_transform(x_train)
x_test=sc.fit_transform(x_test)

/usr/local/lib/python3.9/dist-packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring palette because no hue variable has been ass

x_train.shape
```

```
(8278, 1645)
```

```
def logreg(x_train, x_test, y_train, y_test):
    lr = LogisticRegression(random_state=0)
    lr.fit(x_train,y_train)
    y_lr_tr = lr.predict(x_train)
    print(accuracy_score(y_lr_tr,y_train))
    yPred_lr = lr.predict(x_test)
    print(accuracy_score(yPred_lr,y_test))
    print("***Logistic Regression***")
    print("Confusion_Matrix")
    print(confusion_matrix(y_test,yPred_lr))
    print("Classification Report")
    print(classification_report(y_test,yPred_lr))
```

```
logreg(x_train,x_test,y_train,y_test)
```

```
0.8682048804058952
0.7990338164251207
***Logistic Regression***
Confusion_Matrix
[[820 213]
 [203 834]]
Classification Report
              precision    recall  f1-score   support

     0       0.80        0.79        0.80        1033
     1       0.80        0.80        0.80        1037

 accuracy          0.80          0.80          0.80          2070
 macro avg         0.80          0.80          0.80          2070
 weighted avg      0.80          0.80          0.80          2070
```

```
/usr/local/lib/python3.9/dist-packages/sklearn/linear_model/_logistic.py:458: 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>

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/linear_model.html#logistic-regression)

```
n_iter_i = _check_optimize_result(
```

```
def decisionTree(x_train,x_test,y_train,y_test):
    dtc = DecisionTreeClassifier(criterion="entropy",random_state=0)
    dtc.fit(x_train,y_train)
    y_dt_tr = dtc.predict(x_train)
    print(accuracy_score(y_dt_tr,y_train))
    yPred_dt = dtc.predict(x_test)
    print(accuracy_score(yPred_dt,y_test))
    print("***Decision Tree***")
    print("Confusion_Matrix")
    print(confusion_matrix(y_test,yPred_dt))
    print("Classification Report")
    print(classification_report(y_test,yPred_dt))
```

```
decisionTree(x_train,x_test,y_train,y_test)
```

```
0.9978255617298865
0.793719806763285
***Decision Tree***
Confusion_Matrix
[[672 361]
 [ 66 971]]
```



```

Classification Report
              precision    recall  f1-score   support

      0       0.91      0.65      0.76      1033
      1       0.73      0.94      0.82      1037

 accuracy          0.82
 macro avg          0.82
 weighted avg       0.82

```

```

def RandomForest(x_train,x_test,y_train,y_test):
    rf = RandomForestClassifier(criterion="entropy",n_estimators=10,random_state=0)
    rf.fit(x_train,y_train)
    y_rf_tr = rf.predict(x_train)
    print(accuracy_score(y_rf_tr,y_train))
    yPred_rf = rf.predict(x_test)
    print(accuracy_score(yPred_rf,y_test))
    print("***Random Forest ***")
    print("Confusion Matrix")
    print(confusion_matrix(y_test,yPred_rf))
    print("Classification Report")
    print(classification_report(y_test,yPred_rf))

    /usr/local/lib/python3.7/dist-packages/seaborn/ax/_base.py:1009: UserWarning: Ignoring palette because no hue variable has been assigned
    RandomForest(x_train,x_test,y_train,y_test)

```

```

0.9885237980188452
0.8130434782608695
***Random Forest ***
Confusion Matrix
[[737 296]
 [ 91 946]]
Classification Report
              precision    recall  f1-score   support

      0       0.89      0.71      0.79      1033
      1       0.76      0.91      0.83      1037

 accuracy          0.81
 macro avg          0.83
 weighted avg       0.83

```

```

/usr/local/lib/python3.7/dist-packages/seaborn/ax/_base.py:1009: UserWarning: Ignoring palette because no hue variable has been assigned

```

```

def KNN(x_train,x_test,y_train,y_test):
    knn = KNeighborsClassifier()
    knn.fit(x_train,y_train)
    y_knn_tr = knn.predict(x_train)
    print(accuracy_score(y_knn_tr,y_train))
    yPred_knn = knn.predict(x_test)
    print(accuracy_score(yPred_knn,y_test))
    print("***KNN***")
    print("Confusion Matrix")
    print(confusion_matrix(y_test,yPred_knn))
    print("Classification Report")
    print(classification_report(y_test,yPred_knn))

    /usr/local/lib/python3.7/dist-packages/seaborn/ax/_base.py:1009: UserWarning: Ignoring palette because no hue variable has been assigned

```

```

KNN(x_train,x_test,y_train,y_test)

0.7798985262140614
0.6589371980676328
***KNN***
Confusion Matrix
[[493 540]
 [166 871]]
Classification Report
              precision    recall  f1-score   support

      0       0.75      0.48      0.58      1033
      1       0.62      0.84      0.71      1037

 accuracy          0.66
 macro avg          0.68
 weighted avg       0.68

```

```

/usr/local/lib/python3.7/dist-packages/seaborn/ax/_base.py:1009: UserWarning: Ignoring palette because no hue variable has been assigned

```

```

def svm(x_train,x_test,y_train,y_test):
    svm = SVC(kernel = "linear")

```

```

svm.fit(x_train,y_train)
y_svm_tr = svm.predict(x_train)
print (accuracy_score (y_svm_tr,y_train))
yPred_svm= svm.predict(x_test)
print(accuracy_score(yPred_svm,y_test))
print("***Support Vector Machine***")
print("Confusion_Matrix")
print(confusion_matrix(y_test,yPred_svm))
print("Classification Report")
print(classification_report(y_test,yPred_svm))

func(x=x, y=y, **kwargs)
svm(x_train,x_test,y_train,y_test)

0.8800434887654023
0.7961352657004831
***Support Vector Machine***
Confusion_Matrix
[[756 277]
 [145 892]]
Classification Report
              precision    recall  f1-score   support

     0       0.84      0.73      0.78      1033
     1       0.76      0.86      0.81      1037

 accuracy          0.80
 macro avg          0.80
weighted avg          0.80

func(x=x, y=y, **kwargs)

import keras
from keras.models import Sequential
from keras.layers import Dense

/usr/local/lib/python3.9/dist-packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring palette because no hue variable has been assi
classifier = Sequential()

func(x=x, y=y, **kwargs)

classifier.add(Dense(units=30, activation='relu'))

/usr/local/lib/python3.9/dist-packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no `hue` variable has been assi
classifier.add(Dense(units=30, activation='relu'))

func(x=x, v=v, **kwargs)

classifier.add(Dense(units=1, activation='sigmoid'))

/usr/local/lib/python3.9/dist-packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no `hue` variable has been assi
classifier.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

func(v=v, v=v, **kwargs)

model_history = classifier.fit(x_train, y_train, batch_size=10, validation_split=0.33, epochs=50)

```

```
ann_pred = classifier.predict(x_test)
ann_pred = (ann_pred>0.5)
ann_pred
```

```
65/65 [=====] - 0s 4ms/step
array([[ True],
       [False],
       [ True],
       ...,
       [ True],
       [False],
       [ True]])
```

```
print(accuracy_score(ann_pred,y_test))
print("****ANN Model****")
print("Confusion_Matrix")
print(confusion_matrix(y_test, ann_pred))
print("Classification Report")
print(classification_report(y_test,ann_pred))
```

```
0.8328502415458937
***ANN Model***
Confusion_Matrix
[[896 137]
 [209 828]]
Classification Report
              precision    recall  f1-score   support

      0               0.81        0.87         0.84         1033
      1               0.86        0.80         0.83         1037

 accuracy               0.83                2070
 macro avg              0.83                2070
 weighted avg          0.83                2070
```

[illegible]

```
Predicting on random input
output is: [1]
/usr/local/lib/python3.9/dist-packages/sklearn/linear_model/_logistic.py:458: 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>  
 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/linear_model.html#logistic-regression)

```
n_iter_i = _check_optimize_result(0)
dtc = DecisionTreeClassifier(criterion="entropy", random_state=0)
dtc.fit(x_train,y_train)
print("Predicting on random Input")
dtc_pred_own = dtc.predict(sc.transform([[1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1]]))
print("output is: ",dtc_pred_own)
```

```
Predicting on random Input
output is: [1]
```

```
/usr/local/lib/python3.9/dist-packages/seaborn/axisgrid.py:1600: UserWarning: Ignoring `palette` because no `hue` variable has been assigned
```

```
rf = RandomForestClassifier(criterion="entropy", n_estimators=10, random_state=0)
rf.fit(x_train,y_train)
print("Predicting on rand input")
rf_pred_own=rf.predict(sc.transform([[1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1]]))
print("ouput is: ", rf_pred_own)
```

```
Predicting on rand input
outpu is: [1]
```

```
/usr/local/lib/python3.9/dist-packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no `hue` variable has been assigned
```

```
svc = SVC(kernel = "linear")
svc.fit(x_train,y_train)
print("Predicting on random input")
svm_pred_own = svc.predict(sc.transform([[1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1]]))
print("output is: ", svm_pred_own)
```

```
knn = KNeighborsClassifier()
knn.fit(x_train,y_train)
print("predicting on random input")
knn_pred_own = knn.predict(sc.transform([[1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1]]))
print("output is: ", knn_pred_own)
```

```
predicting on random input
output is: [1]
```

```
/usr/local/lib/python3.9/dist-packages/seaborn/axisgrid.py:1600: UserWarning: Ignoring `palette` because no `hue` variable has been assigned
```

```
ann_classifier = LogisticRegression(solver='lbfgs')
ann_pred_own = ann_classifier.predict(sc.transform([[1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1]]))
print(ann_pred_own)
ann_pred_own = (ann_pred_own>0.5)
print("output is: ", ann_pred_own)
```

```
Predicting on random input
1/1 [[=====] - 0s 25ms/step
[[0.]]
output is: [[False]]
```

```
/usr/local/lib/python3.9/dist-packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring palette because no hue variable has been assigned
```

```
def compareModel(x_train,x_test,y_train,y_test):
    logreg(x_train,x_test,y_train,y_test)
    print('-'*100)
    decisionTree(x_train,x_test,y_train,y_test)
    print('-'*100)
    RandomForest (x_train,x_test,y_train,y_test)
    print('-'*100)
    svm(x_train,x_test,y_train,y_test)
    print('-'*100)
    KNN(x_train,x_test,y_train,y_test)
    print('-'*100)
```

```
/usr/local/lib/python3.9/dist-packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no `hue` variable has been assigned
```

```
compareModel(x_train,x_test,y_train,y_test)
```

```
/usr/local/lib/python3.9/dist-packages/sklearn/linear_model/_logistic.py:458: 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>

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/linear_model.html#logistic-regression)

```
n_iter_i = _check_optimize_result(0)
0.8682048804058952
0.7990338164251207
***Logistic Regression***
Confusion_Matrix
[[820 213]
 [203 834]]
```

```
0.9885237980188452
0.8130434782608695
***Random Forest ***
Confusion Matrix
[[737 296]
 [ 91 946]]
Classification Report
```

	precision	recall	f1-score	support
0	0.89	0.71	0.79	1033
1	0.76	0.91	0.83	1037
accuracy			0.81	2070
macro avg	0.83	0.81	0.81	2070
weighted avg	0.83	0.81	0.81	2070

[illegible]

```

0.9885237980188452
0.8130434782608695
***Random Forest after Hyperparameter tuning***
Confusion_Matric
[[737 296]
 [ 91 946]]
classification Report

```

	precision	recall	f1-score	support
0	0.89	0.71	0.79	1033
1	0.76	0.91	0.83	1037
accuracy			0.81	2070
macro avg	0.83	0.81	0.81	2070
weighted avg	0.83	0.81	0.81	2070

Predicting on random input  
output is: [1]

```

tunc(x=x, y=y, **kwargs)
/usr/local/lib/python3.9/dist-packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no `hue` variable has been assi
func(x=x, y=y, **kwargs)
/usr/local/lib/python3.9/dist-packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no `hue` variable has been assi
func(x=x, y=y, **kwargs)
/usr/local/lib/python3.9/dist-packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no `hue` variable has been assi
func(x=x, y=y, **kwargs)
/usr/local/lib/python3.9/dist-packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no `hue` variable has been assi
func(x=x, y=y, **kwargs)
/usr/local/lib/python3.9/dist-packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no `hue` variable has been assi
func(x=x, y=y, **kwargs)
/usr/local/lib/python3.9/dist-packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no `hue` variable has been assi
func(x=x, y=y, **kwargs)
/usr/local/lib/python3.9/dist-packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no `hue` variable has been assi
func(x=x, y=y, **kwargs)
/usr/local/lib/python3.9/dist-packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no `hue` variable has been assi
func(x=x, y=y, **kwargs)
/usr/local/lib/python3.9/dist-packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no `hue` variable has been assi
func(x=x, y=y, **kwargs)
/usr/local/lib/python3.9/dist-packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no `hue` variable has been assi
func(x=x, y=y, **kwargs)
/usr/local/lib/python3.9/dist-packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no `hue` variable has been assi
func(x=x, y=y, **kwargs)
/usr/local/lib/python3.9/dist-packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no `hue` variable has been assi
func(x=x, y=y, **kwargs)
/usr/local/lib/python3.9/dist-packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no `hue` variable has been assi
func(x=x, y=y, **kwargs)
/usr/local/lib/python3.9/dist-packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no `hue` variable has been assi
func(x=x, y=y, **kwargs)
/usr/local/lib/python3.9/dist-packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no `hue` variable has been assi
func(x=x, y=y, **kwargs)
/usr/local/lib/python3.9/dist-packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no `hue` variable has been assi
func(x=x, y=y, **kwargs)
/usr/local/lib/python3.9/dist-packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no `hue` variable has been assi
func(x=x, y=y, **kwargs)
/usr/local/lib/python3.9/dist-packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no `hue` variable has been assi
func(x=x, y=y, **kwargs)
/usr/local/lib/python3.9/dist-packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no `hue` variable has been assi
func(x=x, y=y, **kwargs)
/usr/local/lib/python3.9/dist-packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no `hue` variable has been assi
func(x=x, y=y, **kwargs)
/usr/local/lib/python3.9/dist-packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no `hue` variable has been assi
func(x=x, y=y, **kwargs)
/usr/local/lib/python3.9/dist-packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no `hue` variable has been assi
func(x=x, y=y, **kwargs)
/usr/local/lib/python3.9/dist-packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no `hue` variable has been assi
func(x=x, y=y, **kwargs)
/usr/local/lib/python3.9/dist-packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no `hue` variable has been assi
func(x=x, y=y, **kwargs)
/usr/local/lib/python3.9/dist-packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no `hue` variable has been assi
func(x=x, y=y, **kwargs)
/usr/local/lib/python3.9/dist-packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no `hue` variable has been assi
func(x=x, y=y, **kwargs)

```

[illegible]

```
<seaborn.axisgrid.PairGrid at 0x7f758f73d790>
```

