

```
#import necessary libraries
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

#import dataset
data = pd.read_csv(r"Churn_Modelling.csv")
data
```

RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember
Automatic document saving has been pending for 2 minutes. Reloading may fix the problem. Save and reload the page. 											
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	1
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1
...
9995	9996	15606229	Obijaku	771	France	Male	39	5	0.00	2	1
9996	9997	15569892	Johnstone	516	France	Male	35	10	57369.61	1	1
9997	9998	15584532	Liu	709	France	Female	36	7	0.00	1	0
9998	9999	15682355	Sabbatini	772	Germany	Male	42	3	75075.31	2	1
9999	10000	15628319	Walker	792	France	Female	28	4	130142.79	1	1

10000 rows × 14 columns

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 14 columns):
 #   Column        Non-Null Count  Dtype  
 ---  --           --           --    
 0   RowNumber    10000 non-null   int64  
 1   CustomerId   10000 non-null   int64  
 2   Surname       10000 non-null   object 
 3   CreditScore  10000 non-null   int64  
 4   Geography     10000 non-null   object 
 5   Gender        10000 non-null   object 
 6   Age           10000 non-null   int64  
 7   Tenure        10000 non-null   int64  
 8   Balance       10000 non-null   float64 
 9   NumOfProducts 10000 non-null   int64  
 10  HasCrCard    10000 non-null   int64  
 11  IsActiveMember 10000 non-null   int64  
 12  EstimatedSalary 10000 non-null   float64 
 13 Exited        10000 non-null   int64  
dtypes: float64(2), int64(9), object(3)
memory usage: 1.1+ MB
```

```
#checking for null values
data.CustomerId=pd.to_numeric(data.CustomerId, errors='coerce')
data.isnull().any()
```

```
RowNumber      False
CustomerId     False
Surname       False
CreditScore    False
Geography     False
Gender        False
Age           False
Tenure         False
Balance        False
NumOfProducts  False
HasCrCard     False
IsActiveMember False
EstimatedSalary False
Exited        False
dtype: bool
```

```
data["CustomerId"].fillna(data["CustomerId"].median(), inplace=True)
data.isnull().sum()
```

```
RowNumber      0
CustomerId     0
Surname       0
CreditScore    0
Geography     0
Gender        0
Age           0
Tenure         0
Balance        0
NumOfProducts  0
HasCrCard     0
```

Automatic document saving has been pending for 2 minutes. Reloading may fix the problem. [Save and reload the page.](#)

```
Exited        0
dtype: int64
```

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
data["RowNumber"] = le.fit_transform(data["RowNumber"])
data["CustomerId"] = le.fit_transform(data["CustomerId"])
data["Surname"] = le.fit_transform(data["Surname"])
data["CreditScore"] = le.fit_transform(data["CreditScore"])
data["Geography"] = le.fit_transform(data["Geography"])
data["Gender"] = le.fit_transform(data["Gender"])
data["Age"] = le.fit_transform(data["Age"])
data["Tenure"] = le.fit_transform(data["Tenure"])
data["Balance"] = le.fit_transform(data["Balance"])
data["NumOfProducts"] = le.fit_transform(data["NumOfProducts"])
data["HasCrCard"] = le.fit_transform(data["HasCrCard"])
data["IsActiveMember"] = le.fit_transform(data["IsActiveMember"])
data["EstimatedSalary"] = le.fit_transform(data["EstimatedSalary"])
data["Exited"] = le.fit_transform(data["Exited"])
```

```
data.head()
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	Estima
0	0	2736	1115	228	0	0	24	2	0	0	1	1	1
1	1	3258	1177	217	2	0	23	1	743	0	0	0	1
2	2	2104	2040	111	0	0	24	8	5793	2	1	0	0
3	3	5435	289	308	0	0	21	1	0	1	0	0	0
4	4	6899	1822	459	2	0	25	2	3696	0	1	1	1

```
x=data.iloc[:,0:14].values
y=data.iloc[:,13:14].values
```

```
x
```

```
array([[ 0, 2736, 1115, ... , 1, 5068, 1],
 [ 1, 3258, 1177, ... , 1, 5639, 0],
 [ 2, 2104, 2040, ... , 0, 5707, 1],
 ... ,
 [9997, 717, 1570, ... , 1, 2062, 1],
```

```
[9998, 4656, 2345, ...], 0, 4639, 1],
[9999, 2497, 2751, ...], 0, 1878, 0]])
```

y

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

```
from sklearn.preprocessing import OneHotEncoder
one = OneHotEncoder()
a= one.fit_transform(x[:,0:1]).toarray()
b= one.fit_transform(x[:,1:2]).toarray()
c= one.fit_transform(x[:,2:3]).toarray()
d= one.fit_transform(x[:,3:4]).toarray()
e= one.fit_transform(x[:,4:5]).toarray()
f= one.fit_transform(x[:,5:6]).toarray()
g= one.fit_transform(x[:,6:7]).toarray()
h= one.fit_transform(x[:,7:8]).toarray()
i= one.fit_transform(x[:,8:9]).toarray()
j= one.fit_transform(x[:,9:10]).toarray()
x=np.delete(x, [0,1,2,3,4,5,6,7,8,9], axis=1)
x=np.concatenate((a,b,c,d,e,f,g,h,i,j,x), axis=1)
```

Automatic document saving has been pending for 2 minutes. Reloading may fix the problem. [Save and reload the page.](#) ×

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

x_resample

```
array([[1.00000000e+00, 0.00000000e+00, 0.00000000e+00, ...,
       1.00000000e+00, 5.06800000e+03, 1.00000000e+00],
       [0.00000000e+00, 1.00000000e+00, 0.00000000e+00, ...,
       1.00000000e+00, 5.63900000e+03, 0.00000000e+00],
       [0.00000000e+00, 0.00000000e+00, 1.00000000e+00, ...,
       0.00000000e+00, 5.70700000e+03, 1.00000000e+00],
       ...,
       [0.00000000e+00, 0.00000000e+00, 0.00000000e+00, ...,
       0.00000000e+00, 9.87599651e+03, 1.00000000e+00],
       [0.00000000e+00, 0.00000000e+00, 0.00000000e+00, ...,
       5.98737695e-01, 6.03421010e+03, 1.00000000e+00],
       [0.00000000e+00, 0.00000000e+00, 0.00000000e+00, ...,
       0.00000000e+00, 9.51229892e+03, 1.00000000e+00]])
```

y_resample

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

data.describe()

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProd
count	10000.00000	10000.00000	10000.00000	10000.00000	10000.00000	10000.00000	10000.00000	10000.00000	10000.00000	10000.00C
mean	4999.50000	4999.50000	1507.774200	259.584600	0.746300	0.545700	20.920600	5.012800	2036.788100	0.53
std	2886.89568	2886.89568	846.204311	96.496107	0.827529	0.497932	10.482065	2.892174	2125.232536	0.58
min	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00C
25%	2499.75000	2499.75000	773.75000	193.00000	0.00000	0.00000	14.00000	3.00000	0.00000	0.00C
50%	4999.50000	4999.50000	1542.00000	261.00000	0.00000	1.00000	19.00000	5.00000	1383.50000	0.00C
75%	7499.25000	7499.25000	2238.25000	327.00000	1.00000	1.00000	26.00000	7.00000	3882.25000	1.0C
max	9999.00000	9999.00000	2931.00000	459.00000	2.00000	1.00000	69.00000	10.00000	6381.00000	3.0C

```
plt.figure(figsize=(12,5))
plt.subplot(1,2,1)
```

```

sns.distplot(data["Balance"])
plt.subplot(1,2,2)
sns.distplot(data["Tenure"])

<ipython-input-21-40ad5a6b3819>: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["Balance"])
<ipython-input-21-40ad5a6b3819>:5: 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).

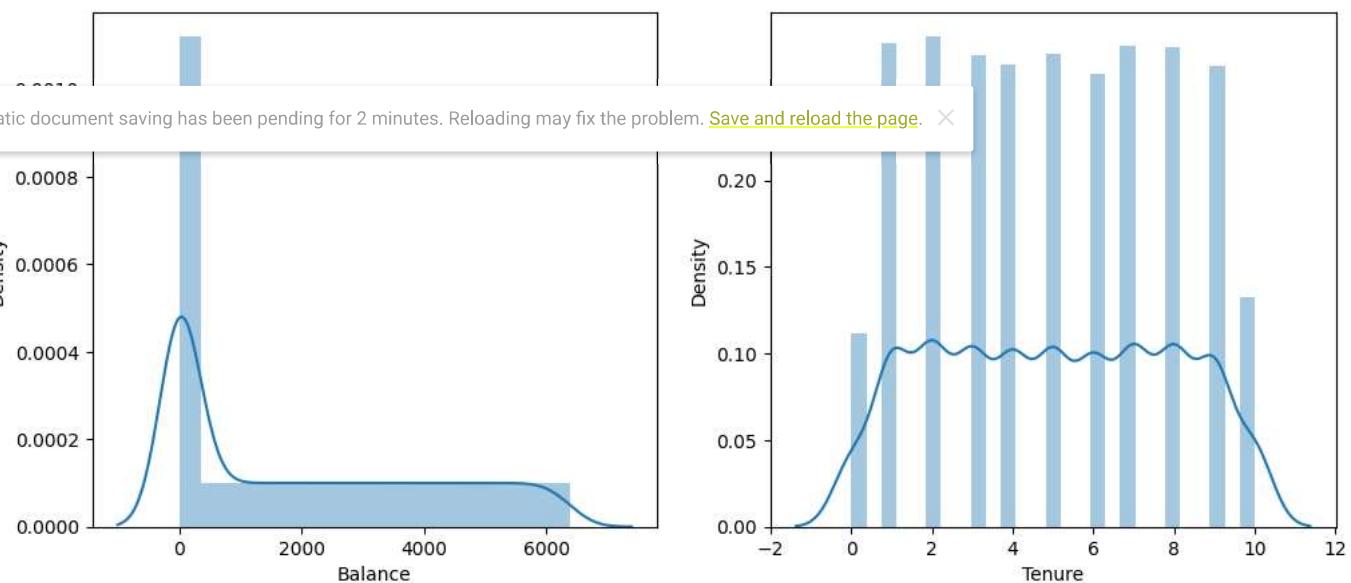
```

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

```

sns.distplot(data["Tenure"])
<Axes: xlabel='Tenure', ylabel='Density'>

```



```

plt.figure(figsize=(12,5))
plt.subplot(1,2,1)
sns.countplot(data["Tenure"])
plt.subplot(1,2,1)
sns.countplot(data["Balance"])

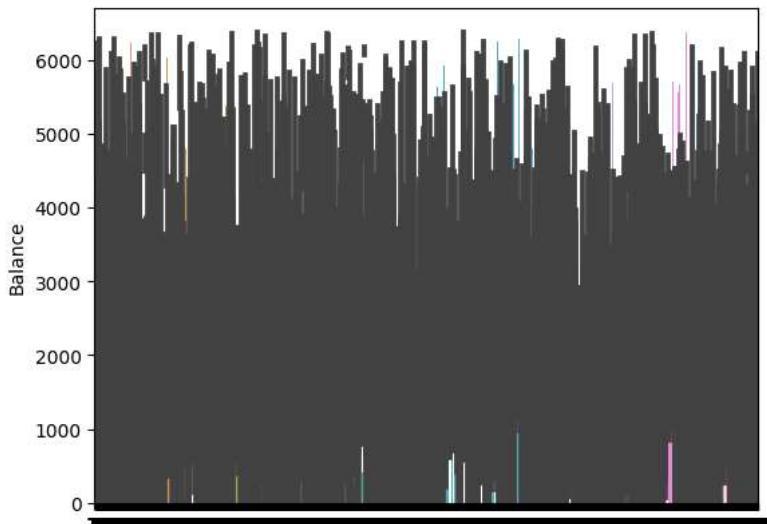
```

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



```
sns.barplot(x="Surname", y="Balance", data=data)
```

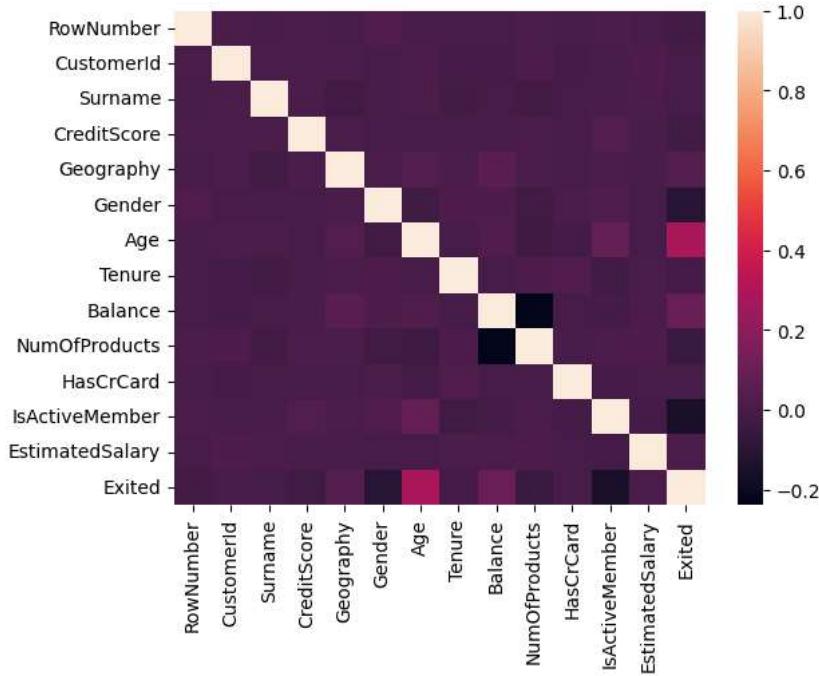
```
<Axes: xlabel='Surname', ylabel='Balance'>
```



Automatic document saving has been pending for 2 minutes. Reloading may fix the problem. [Save and reload the page.](#) ×

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

```
<Axes: >
```



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

Automatic document saving has been pending for 2 minutes. Reloading may fix the problem. [Save and reload the page.](#)

4/12/23, 3:27 PM

Untitled1.ipynb - Colaboratory

```
/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  
x_train.ndim  
  
2  
+tunc(x=x, y=y, **kwargs)
```

```
x_test.ndim  
  
2  
+tunc(x=x, y=y, **kwargs)
```

```
x_train=pd.DataFrame(x_train)  
x_test=pd.DataFrame(x_test)  
+tunc(x=x, y=y, **kwargs)
```

```
x_train
```

	0	1	2	3	4	5	6	7	8	9	...	29858	29859	29860	29861	29862	29863	29864	29865	29866
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	1.000000	0.000000	0.000000	0.0	0.930159	0.069841	6636.768249
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	1.000000	0.000000	0.000000	0.0	0.000000	0.000000	7113.000000
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	1.000000	0.000000	0.000000	0.0	0.043995	1.000000	9852.296088
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	1.000000	0.000000	0.000000	0.0	1.000000	0.000000	5534.000000
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	1.000000	0.000000	0.000000	0.0	0.159173	0.000000	4274.522482
...	
12735	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.209732	0.790268	0.000000	0.0	0.209732	0.000000	6568.951340
12736	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.000000	1.000000	0.000000	0.0	1.000000	0.000000	9083.000000
12737	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.000000	1.000000	0.000000	0.0	1.000000	1.000000	7470.000000
12738	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.123083	0.000000	0.876917	0.0	0.123083	0.000000	391.107750
12739	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	1.000000	0.000000	0.000000	0.0	1.000000	0.000000	5952.000000

```
12740 rows × 29868 columns
```

```
from sklearn.preprocessing import StandardScaler  
sc=StandardScaler()  
x_train = sc.fit_transform(x_train)  
x_test = sc.fit_transform(x_test)  
+tunc(x=x, v=v, **kwargs)
```

```
x_train.shape
```

```
/usr/local/lib/python3.9/dist-packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring palette because no 'hue' variable has been ass:  
#importing and building the Decision tree model  
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))
```

```
/usr/local/lib/python3.9/dist-packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring palette because no 'hue' variable has been ass  
#printing the train accuracy and test accuracy respectively  
logreg(x_train,x_test,y_train,y_test)
```

```
/usr/local/lib/python3.9/dist-packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring palette because no 'hue' variable has been ass  
#importing and building the Decision tree model  
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))

#printing the train accuracy and test accuracy respectively
decisionTree(x_train,x_test,y_train,y_test)

#importing and building the random forest model
def RandomForest(x_tarin,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))

```

Automatic document saving has been pending for 2 minutes. Reloading may fix the problem. [Save and reload the page.](#)

```

#importing and building the KNN model
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))

```

```

#printing the train accuracy and test accuracy respectively
KNN(x_train,x_test,y_train,y_test)

```

```

#importing and building the random forest model
def svm(x_tarin, 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))

```

```

#printing the train accuracy and test accuracy respectively
svm(x_train,x_test,y_train,y_test)

```

```

# Importing the Keras libraries and packages
import keras
from keras.models import Sequential
from keras.layers import Dense

] # Initialising the ANN
classifier Sequential()


```

```
[ ] # Adding the input layer and the first hidden layer
classifier.add(Dense (units=30, activation='relu', input_dim=40))

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

# Adding the second hidden layer
classifier.add(Dense (units=30, activation='relu'))

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

# Adding the output layer
classifier.add(Dense (units=1, activation='sigmoid'))

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

# Compiling the ANN
classifier.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

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

#fitting the ANN to the training set
model_history=classifier.fit(x_train,y_train,batch_size=10,validation_split=0.33,epochs=200)

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

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

/usr/local/lib/python3.9/dist-packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no `hue` variable has been assi
print(accuracy_score (ann_pred,y_test))
print("****ANN Model****")
print("Confusion_Matrix")
print(confusion_matrix(y_test, ann_pred))
print("Classification Report")

Automatic document saving has been pending for 2 minutes. Reloading may fix the problem. Save and reload the page. × Because no hue variable has been assi

#testing on random input values
lr = LogisticRegression(random_state=0)
lr.fit(x_train,y_train)
print("Predicting on random input")
lr_pred_own = lr.predict(sc.transform([[0,0,1,1,0,0,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,1,0,0,456,1,0,3245,4567]]))
print("output is: ",lr_pred_own)

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

#testing on random input values
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([[0,0,1,1,0,0,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,1,0,0,456,1,0,3245,4567]]))
print("output is: ",dtc_pred_own)

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

#testing on random input values
rf = RandomForestClassifier (criterion="entropy",n_estimators=10, random_state=0)
rf.fit(x_train,y_train)
print("Predicting on random input")
rf_pred_own = rf.predict(sc.transform([[0,0,1,1,0,0,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,1,0,0,456,1,0,3245,4567]]))
print("output is: ",rf_pred_own)

/usr/local/lib/python3.9/dist-packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring palette because no hue variable has been assi
#testing on random input values
Svc = SVC (kernel = "linear")
svc.fit(x_train,y_train)
print("Predicting on random input")
svm_pred_own = svc.predict(sc.transform([[0,0,1,1,0,0,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,1,0,0,456,1,0,3245,4567]]))
print("output is: ", svm_pred_own)

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

#testing on random input values
knn = KNeighborsClassifier()
knn.fit(x_train,y_train)
print("Predicting on random input")
knn_pred_own = knn.predict(sc.transform([[0,0,1,1,0,0,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,1,0,0,456,1,0,3245,4567]]))
print("output is: ",knn_pred_own)

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

#testing on random input values
print("Predicting on random input")
ann_pred_own = classifier.predict(sc.transform([[0,0,1,1,0,0,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,1,0,0,456,1,0,3245,4567]]))
print (ann_pred_own)
ann_pred_own = (ann_pred_own>0.5)
print("output is: ", ann_pred_own)

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

```

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)

    func(v=v v=v **kwargs)
compareModel(x_train,x_test,y_train,y_test)

/usr/local/lib/python3.9/dist-packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no `hue` variable has been assi
y_rf=model.predict(x_train)
print(accuracy_score (y_rf,y_train))
yPred_rfcv = model.predict(x_test) =
print(accuracy_score (yPred_rfcv,y_test))
print("****Random Forest after Hyperparameter tuning***")
print("Confusion_Matrix")
print(confusion_matrix(y_test,yPred_rfcv))
print("Classification Report")
print(classification_report (y_test,yPred_rfcv))
print("Predicting on random input")
rfcv_pred_own = model.predict(sc.transform([[0,0,1,1,0,0,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,456,1,0,3245,4567]]))
print("output is: ",rfcv_pred_own)

```

Automatic document saving has been pending for 2 minutes. Reloading may fix the problem. [Save and reload the page](#). X because no `hue` variable has been assi

```

/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)

```

<seaborn.axisgrid.PairGrid at 0x7fd334631130>

