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

**import** seaborn **as** sb

df **=** pd**.**read\_csv("Churn\_Modelling.csv")

df**.**shape

df**.**columns

df**.**head()

*#input data*

x **=** df[['CreditScore', 'Age', 'Tenure',

'Balance', 'NumOfProducts',

'HasCrCard', 'IsActiveMember', 'EstimatedSalary']]

*#output data*

y **=** df['Exited']

x

sb**.**countplot(x **=** y)

y**.**value\_counts()

*# Balancing the dataset*

**!**pip install imbalanced-learn

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

ros **=** RandomOverSampler(random\_state**=**0)

x\_res, y\_res **=** ros**.**fit\_resample(x,y)

y\_res**.**value\_counts()

*# Normalize*

**from** sklearn.preprocessing **import** StandardScaler

scaler **=** StandardScaler()

x\_scaled **=** scaler**.**fit\_transform(x)

x\_scaled

*# Cross-validation*

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

x\_train,x\_test,y\_train,y\_test **=** train\_test\_split(

x\_scaled,y,random\_state**=**0, test\_size**=**0.25)

x**.**shape

x\_test**.**shape

x\_train**.**shape

*# Neural Network*

**from** sklearn.neural\_network **import** MLPClassifier

ann **=** MLPClassifier(hidden\_layer\_sizes **=** (100,100,100),

random\_state**=**0, max\_iter**=**100,

activation**=**'relu')

ann**.**fit(x\_train,y\_train)

y\_pred **=** ann**.**predict(x\_test)

**from** sklearn.metrics **import** accuracy\_score, ConfusionMatrixDisplay

**from** sklearn.metrics **import** accuracy\_score, classification\_report

ConfusionMatrixDisplay**.**from\_predictions(y\_test, y\_pred)

accuracy\_score(y\_test, y\_pred)

print(classification\_report(y\_test, y\_pred))

*#problem is 0.45*

*#identify point of improvement and and implement the same*

*# to improve this we need to balance the dataset first see above i will use imblearn*

*#balance sampler*