**Download the dataset from Kaggle.com**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Steps in NN construction**

1. **Import libraries**.
2. **Load dataset** from the CSV file.
3. **Preprocess data** (drop irrelevant columns, encode categorical variables).
4. **Split data** into training and testing sets.
5. **Scale features** to improve neural network performance.
6. **Build a neural network model**.
7. **Compile the model** with a loss function and optimizer.
8. **Train the model** on the training data.
9. **Make predictions** on the test data.
10. **Evaluate the model** using accuracy and confusion matrix.

**Step 1**

* numpy: For numerical operations like handling arrays and matrices.
* pandas: For loading and manipulating data (typically in CSV files).
* tensorflow.keras: For creating and training neural networks.
* train\_test\_split: From sklearn, used to split the data into training and testing sets.
* StandardScaler: From sklearn, used to scale numerical features.
* accuracy\_score and confusion\_matrix: From sklearn, used to evaluate model performance.

Step 2:

df: This is the DataFrame that contains our dataset. pd.read\_csv reads the CSV file into a table-like structure.

**CODE:**

**# step 1: importing libraries**

import numpy as np # for numerical calculation

import pandas as pd # for reading and manipulating datasets

import tensorflow

from tensorflow.keras.models import Sequential

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

from sklearn.preprocessing import StandardScaler

from sklearn.metrics import accuracy\_score,confusion\_matrix

**# step 2: loading dataset**

df=pd.read\_csv('d:/Churn\_modelling.csv')

print(df.head()) # to view first few rows of dataset

**# step 3: preprocessing the data**

print(df.columns) # just to display column heading

df = df.replace({True: 1, False: 0})

df=df.drop(['CustomerId','Surname'],axis=1)

df=pd.get\_dummies(df,drop\_first=True)

x=df.drop('Exited',axis=1).values

y=df['Exited'].values

**# Step 4: Splitting the data into Training and Testing**

x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,test\_size=0.2,random\_state=42)

**# Step 5: Scaling the data**

scaler=StandardScaler()

X\_train=scaler.fit\_transform(x\_train)

X\_test=scaler.transform(x\_test)

**# Step 6: Building a neural network**

from tensorflow.keras.layers import Dense

model=Sequential()

model.add(Dense(64,input\_dim=x\_train.shape[1],activation='relu'))

model.add(Dense(32,activation='relu'))

model.add(Dense(1,activation='sigmoid'))

**# Step 7: compile**

model.compile(loss='binary\_crossentropy',optimizer='adam',metrics=['accuracy'])

**# Step 8: Training**

model.fit(x\_train,y\_train,epochs=10)

**# Step 9: Making Prediction**

y\_pred=model.predict(x\_test)>0.5

print(y\_pred)

**# Step 10: Evaluating the model**

from sklearn.metrics import accuracy\_score, confusion\_matrix

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

c\_matrix=confusion\_matrix(y\_test,y\_pred)

print(f'Accuracy:{accuracy:.2f}')

print('Confusion Matrix:')

print(c\_matrix)