churn_nn_colab.py

```
# -*- coding: utf-8 -*-
"""churn_NN.ipynb

Automatically generated by Colaboratory.

Original file is located at
https://colab.research.google.com/drive/1RB4zjW8MhMLA9l2by25QyhdC5-9g3AHP
"""

import numpy as np
import pandas as pd
import math
import matplotlib.pyplot as plt
import seaborn as sns

from google.colab import files
uploded = files.upload()

import io
df = pd.read_csv(io.BytesIO(uploded['Churn_Modelling.csv']))
df.shape

df.drop(['CustomerId','RowNumber','Surname'], axis = 'columns', inplace =True)

df.isna().sum()



df.dtypes

df['Geography'].unique()

#one hot encoding
df = pd.get_dummies(data = df, columns=['Geography'])
df.dtypes

df['Gender'].unique()

df['Gender'].replace(['Male', 'Female'],[1, 0], inplace= True)

df['Exited'].value_counts()

#separate outcome or target col
X = df.drop(['Exited'], axis=1)
y = df['Exited']

from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
X_train,X_test,y_train,y_test =
train_test_split(X,y,test_size=0.2,random_state=0)
```

```
from sklearn.preprocessing import StandardScaler
#
# feature scaling
#
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

import tensorflow as tf
from tensorflow import keras

model = keras.Sequential([
keras.layers.Dense(12, input_shape=(12,),activation='relu'),
keras.layers.Dense(15, activation='relu'),
keras.layers.Dense(1, activation='sigmoid')
])

model.compile(optimizer='adam',
loss='binary_crossentropy',
metrics=['accuracy'])

model.fit(X_train, y_train, epochs=100)

model.evaluate(X_test, y_test)

yp = model.predict(X_test)

from sklearn.metrics import confusion_matrix , classification_report
print(classification_report(y_test,yp))

y_pred = []
for element in yp:
if element > 0.5:
y_pred.append(1)
else:
y_pred.append(0)

print(classification_report(y_test,y_pred))

cm = tf.math.confusion_matrix(labels=y_test,predictions=y_pred)

cm

tf.math.confusion_matrix(labels=y_test,predictions=y_pred)
```