

ĐIỆN TOÁN ĐÁM MÂY / IS402.N11.HTCL

GVHD: THS. HÀ LÊ HOÀI TRUNG

Bank Customer Churn Prediction using ANN



GROUP

THÀNH VIÊN NHÓM

NGUYỄN CAO KHOA
19522177



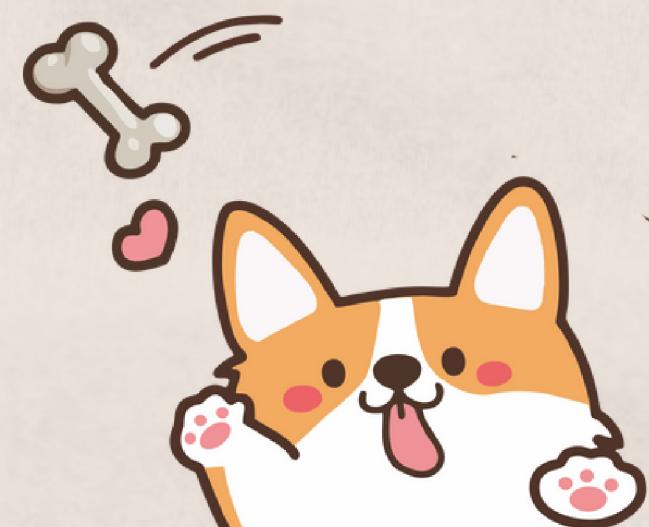
TRẦN NHẬT TÂN
19521825



HỒ TRỌNG KHANG
19521463



HUỲNH QUỐC KHÁNH
19521463



CONTENTS

1

Lý do chọn đề tài

2

Mô tả dữ liệu gốc

3

Tiền xử lý dữ liệu

4

Thư viện Tensorflow

5

Algorithm models

6

Kết luận

LÝ DO CHỌN ĐÈ TĀI ??



MÔ TẢ DỮ LIỆU GỐC

TẬP DỮ LIỆU

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	
1	RowNum	Customer	Surname	CreditScore	Geograph	Gender	Age	Tenure	Balance	NumOfPr	HasCrCarc	IsActiveM	Estimated	Exited	
2	1	15634602	Hargrave	619	France	Female	42	2	0	1	1	1	101348.9	1	
3	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.6	0	
4	3	15619304	Onio	502	France	Female	42	8	159660.8	3	1	0	113931.6	1	
5	4	15701354	Boni	699	France	Female	39	1	0	2	0	0	93826.63	0	
6	5	15737888	Mitchell	850	Spain	Female	43	2	125510.8	1	1	1	79084.1	0	
7	6	15574012	Chu	645	Spain	Male	44	8	113755.8	2	1	0	149756.7	1	
8	7	15592531	Bartlett	822	France	Male	50	7	0	2	1	1	10062.8	0	
9	8	15656148	Obinna	376	Germany	Female	29	4	115046.7	4	1	0	119346.9	1	
10	9	15792365	He	501	France	Male	44	4	142051.1	2	0	1	74940.5	0	
11	10	15592389	H?	684	France	Male	27	2	134603.9	1	1	1	71725.73	0	
12	11	15767821	Bearce	528	France	Male	31	6	102016.7	2	0	0	80181.12	0	
13	12	15737173	Andrews	497	Spain	Male	24	3	0	2	1	0	76390.01	0	
14	13	15632264	Kay	476	France	Female	34	10	0	2	1	0	26260.98	0	
15	14	15691483	Chin	549	France	Female	25	5	0	2	0	0	190857.8	0	
16	15	15600882	Scott	635	Spain	Female	35	7	0	2	1	1	65951.65	0	
17	16	15643966	Goforth	616	Germany	Male	45	3	143129.4	2	0	1	64327.26	0	
18	17	15737452	Romeo	653	Germany	Male	58	1	132602.9	1	1	0	5097.67	1	
19	18	15788218	Henderso	549	Spain	Female	24	9	0	2	1	1	14406.41	0	
20	19	15661507	Muldrow	587	Spain	Male	45	6	0	1	0	0	158684.8	0	
21	20	15568982	Hao	726	France	Female	24	6	0	2	1	1	54724.03	0	
22	21	15577657	McDonald	732	France	Male	41	8	0	2	1	1	170886.2	0	
23	22	15597945	Dellucci	636	Spain	Female	32	8	0	2	1	0	138555.5	0	
24	23	15699309	Gerasimo	510	Spain	Female	38	4	0	1	1	0	118913.5	1	
25	24	15725737	Mosman	669	France	Male	46	3	0	2	0	1	8487.75	0	
26	25	15625047	Yen	846	France	Female	38	5	0	1	1	1	187616.2	0	
27	26	15738191	Maclean	577	France	Male	25	3	0	2	0	1	124508.3	0	
28	27	15736816	Young	756	Germany	Male	36	2	136815.6	1	1	1	170042	0	
29	28	15700772	Nebechi	571	France	Male	44	9	0	2	0	0	38433.35	0	
30	29	15728693	McWillian	574	Germany	Female	43	3	141349.4	1	1	1	100187.4	0	
31	30	15656300	Lucciano	411	France	Male	29	0	59697.17	2	1	1	53483.21	0	
32	31	15589475	Azikiwe	591	Spain	Female	39	3	0	3	1	0	140469.4	1	
33	32	15706552	Odinakach	533	France	Male	36	7	85311.7	1	0	1	156731.9	0	
34	33	15750181	Sanderson	553	Germany	Male	41	9	110112.5	2	0	0	81898.81	0	

NHÓM THUỘC TÍNH: THÔNG TIN CỦA KHÁCH HÀNG

Tên cột	Ý nghĩa	Kiểu dữ liệu	Miền giá trị
CustomerId	Id của khách hàng trong tập dữ liệu.	Numeric	350-850
Surname	Họ của khách hàng.	String	
Geography	Tỉnh thành hoặc khu vực khách hàng sinh sống.	String	3
Gender	Giới tính.	String	Female/Male
Age	Độ tuổi.	Numeric	18-92
EstimatedSalary	Mức lương ước tính.	Numeric	11.6-200.000

NHÓM THUỘC TÍNH: DỮ LIỆU NGÂN HÀNG CỦA KHÁCH HÀNG

Tên cột	Ý nghĩa	Kiểu dữ liệu	Miền giá trị
CreditScore	Điểm tín dụng của khách hàng.	Numeric	350-850
Tenure	Số năm khách hàng sử dụng dịch vụ.	Numeric	0-10
Balance	Tỉnh thành hoặc khu vực khách hàng sinh sống.	Numeric	0-251.000
NumOfProducts	Số lượng sản phẩm tiện ích của ngân hàng mà khách hàng đang sử dụng.	Numeric	1-4
HasCrCard	Khách hàng có thẻ tín dụng không ?	Binary	1: có 0: không
IsActiveMember	Khách hàng có còn hoạt động không ?	Binary	1: có 0: không
Exited	Khách hàng rời bỏ ?	Binary	1: có 0: không

NHÓM THUỘC TÍNH: KHÁC

Tên cột	Ý nghĩa	Kiểu dữ liệu	Miền giá trị
RowIndex	Số lượng dữ liệu trong tập dữ liệu	Numeric	1-10.000

TIỀN XỬ LÝ DỮ LIỆU

Finance was always brought back to life anywhere



Kết nối Drive và Import thư viện

Kết nối Drive

```
✓ 21s   from google.colab import drive  
       drive.mount('/gdrive')  
  
→ Mounted at /gdrive
```

Import các thư viện

```
[ ] import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
from sklearn.preprocessing import MinMaxScaler  
from sklearn.model_selection import train_test_split  
import tensorflow as tf  
from tensorflow import keras  
import seaborn as sn  
from sklearn.metrics import confusion_matrix,classification_report
```

Import dataset

```
[15] df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/Churn Modeling.csv')
df
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	101348.88	1
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	113931.57	1
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	93826.63	0
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	0
...
9995	9996	15606229	Obijaku	771	France	Male	39	5	0.00	2	1	0	96270.64	0
9996	9997	15569892	Johnstone	516	France	Male	35	10	57369.61	1	1	1	101699.77	0
9997	9998	15584532	Liu	709	France	Female	36	7	0.00	1	0	1	42085.58	1
9998	9999	15682355	Sabbatini	772	Germany	Male	42	3	75075.31	2	1	0	92888.52	1
9999	10000	15628319	Walker	792	France	Female	28	4	130142.79	1	1	0	38190.78	0

10000 rows × 14 columns



Xóa các cột không cần thiết

Xóa cột 'RowNumber', 'CustomerId', 'Surname' ;

```
[16] df.drop(['RowNumber', 'CustomerId', 'Surname'], axis=1, inplace = True)  
df.shape  
  
(10000, 11)
```

Kiểm tra dữ liệu null

```
[22] df.isnull().values.any()
```

False

Kiểm tra kiểu dữ liệu các thuộc tính

```
[23] df.dtypes
```

CreditScore	int64
Geography	object
Gender	object
Age	int64
Tenure	int64
Balance	float64
NumOfProducts	int64
HasCrCard	int64
IsActiveMember	int64
EstimatedSalary	float64
Exited	int64
dtype:	object

Kiểm tra Oulier của các thuộc tính numeric

On "Job" column

```
encoder = LabelEncoder()
df_filtered['job'] = encoder.fit_transform(df_filtered['job'])
job = {index : label for index, label in enumerate(encoder.classes_)}
job

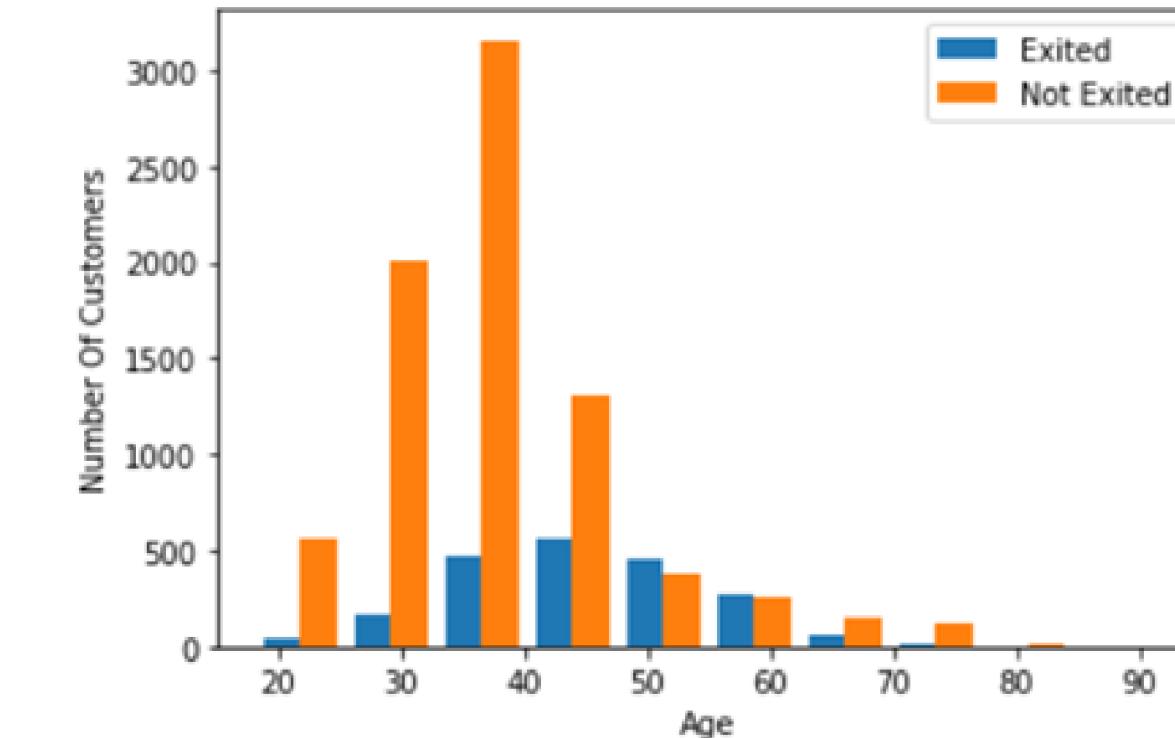
{0: 'admin.',
 1: 'blue-collar',
 2: 'entrepreneur',
 3: 'housemaid',
 4: 'management',
 5: 'retired',
 6: 'self-employed',
 7: 'services',
 8: 'student',
 9: 'technician',
10: 'unemployed'}
```

Trực quan hóa dữ liệu

Thuộc tính 'Age' và 'Exited'

```
[1]: Age_Exited = df[df.Exited == 1].Age  
Age_NotExited = df[df.Exited == 0].Age  
  
plt.xlabel("Age")  
plt.ylabel("Number Of Customers")  
plt.hist([Age_Exited,Age_NotExited],label=['Exited','Not Exited'])  
plt.legend()
```

```
[2]: /usr/local/lib/python3.8/dist-packages/numpy/core/fromnumeric.py:3208:  
      return asarray(a).size  
/usr/local/lib/python3.8/dist-packages/matplotlib/cbook/__init__.py:1:  
      X = np.atleast_1d(X.T if isinstance(X, np.ndarray) else np.asarray(X))  
<matplotlib.legend.Legend at 0x7fa4b6916ee0>
```

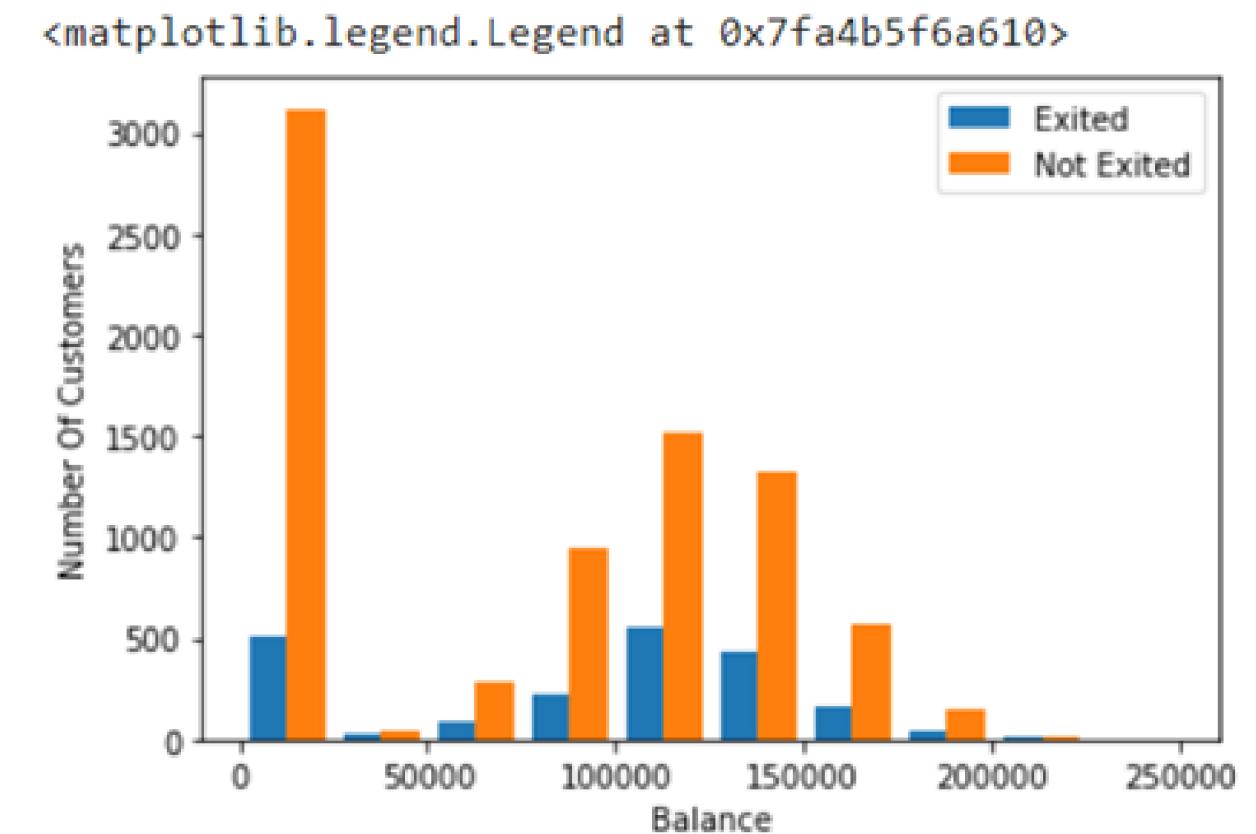


Trực quan hóa dữ liệu

Thuộc tính
'Balance' và
'Exited'

```
[27] Balance_Exited = df[df.Exited == 1].Balance
    Balance_NotExited = df[df.Exited == 0].Balance

    plt.xlabel("Balance")
    plt.ylabel("Number Of Customers")
    plt.hist([Balance_Exited,Balance_NotExited],label=['Exited','Not Exited'])
    plt.legend()
```



Trực quan hóa dữ liệu

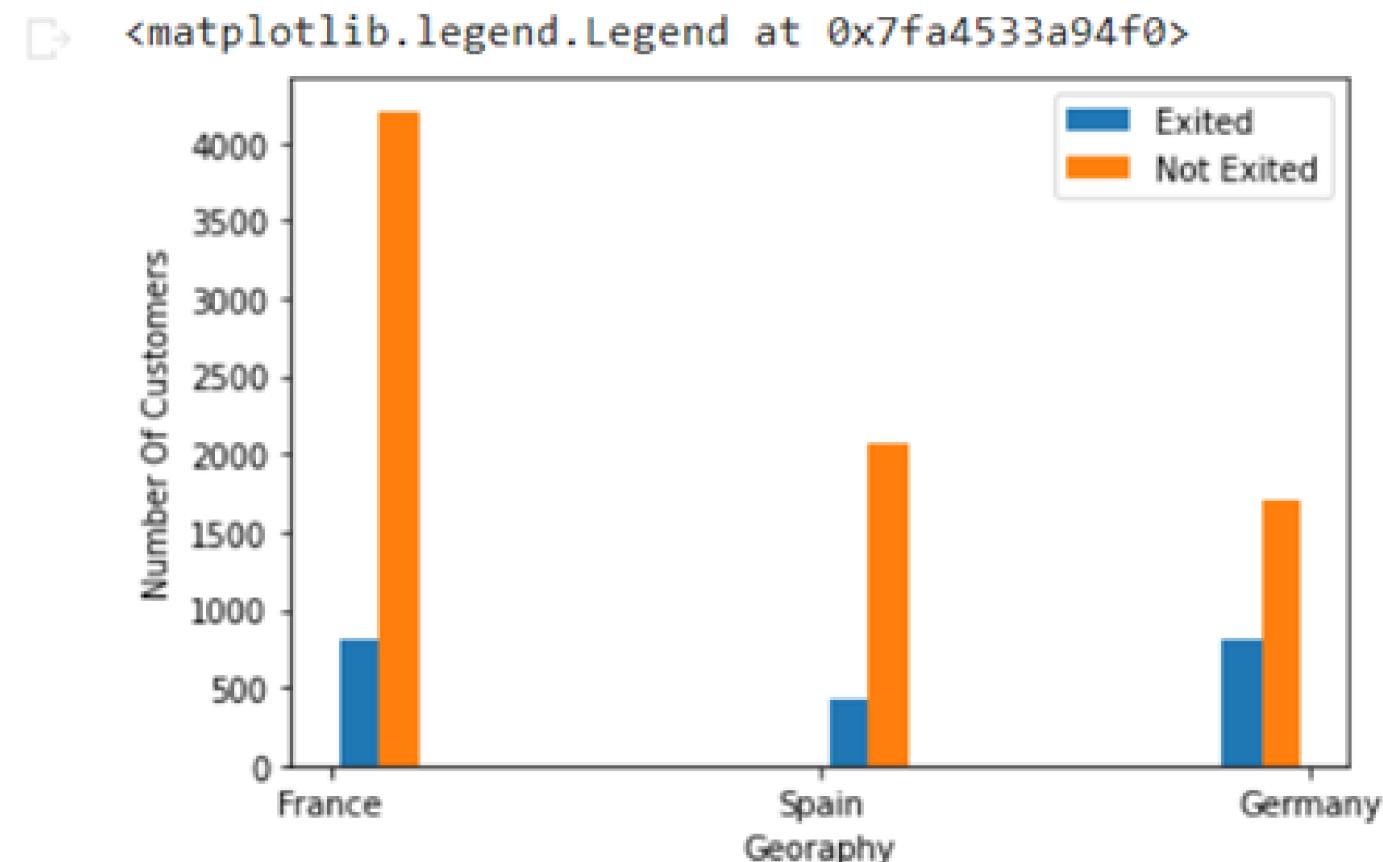
Thuộc tính
'Gender' và
'Exited'



Trực quan hóa dữ liệu

Thuộc tính
'Georaphy' và
'Exited'

```
[43] Geography_Exited = df[df.Exited == 1].Geography  
Geography_NotExited = df[df.Exited == 0].Geography  
  
plt.xlabel("Geography")  
plt.ylabel("Number Of Customers")  
plt.hist([Geography_Exited,Geography_NotExited],label=['Exited','Not Exited'])  
plt.legend()
```



Chuyển đổi data dạng chữ sang số

```
[ ] df2 = pd.get_dummies(data = df,columns=[ 'Geography' , 'Gender' ])
df2.head()
```

#	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited	Geography_France	Geography_Germany	Geography_Spain	Gender_Female	Gender_Male
2	0.00	1	1	1	101348.88	1	1	0	0	1	0
1	83807.86	1	0	1	112542.58	0	0	0	1	1	0
8	159660.80	3	1	0	113931.57	1	1	0	0	1	0
1	0.00	2	0	0	93826.63	0	1	0	0	1	0
2	125510.82	1	1	1	79084.10	0	0	0	1	1	0

Chuyển đổi data

MinMaxScaler

```
from sklearn.preprocessing import MinMaxScaler  
scaler = MinMaxScaler()  
df2[['CreditScore','EstimatedSalary',  
     'Balance','Age']] = scaler.fit_transform(  
df2[['CreditScore','EstimatedSalary',  
     'Balance','Age']])
```

Kết quả

- *Loại bỏ các cột thừa*
- *Loại bỏ các dữ liệu null*
- *Chuyển đổi dữ liệu data để tiến hành mining.*

THƯ VIỆN TENSORFLOW

... was once brought back to life anywhere

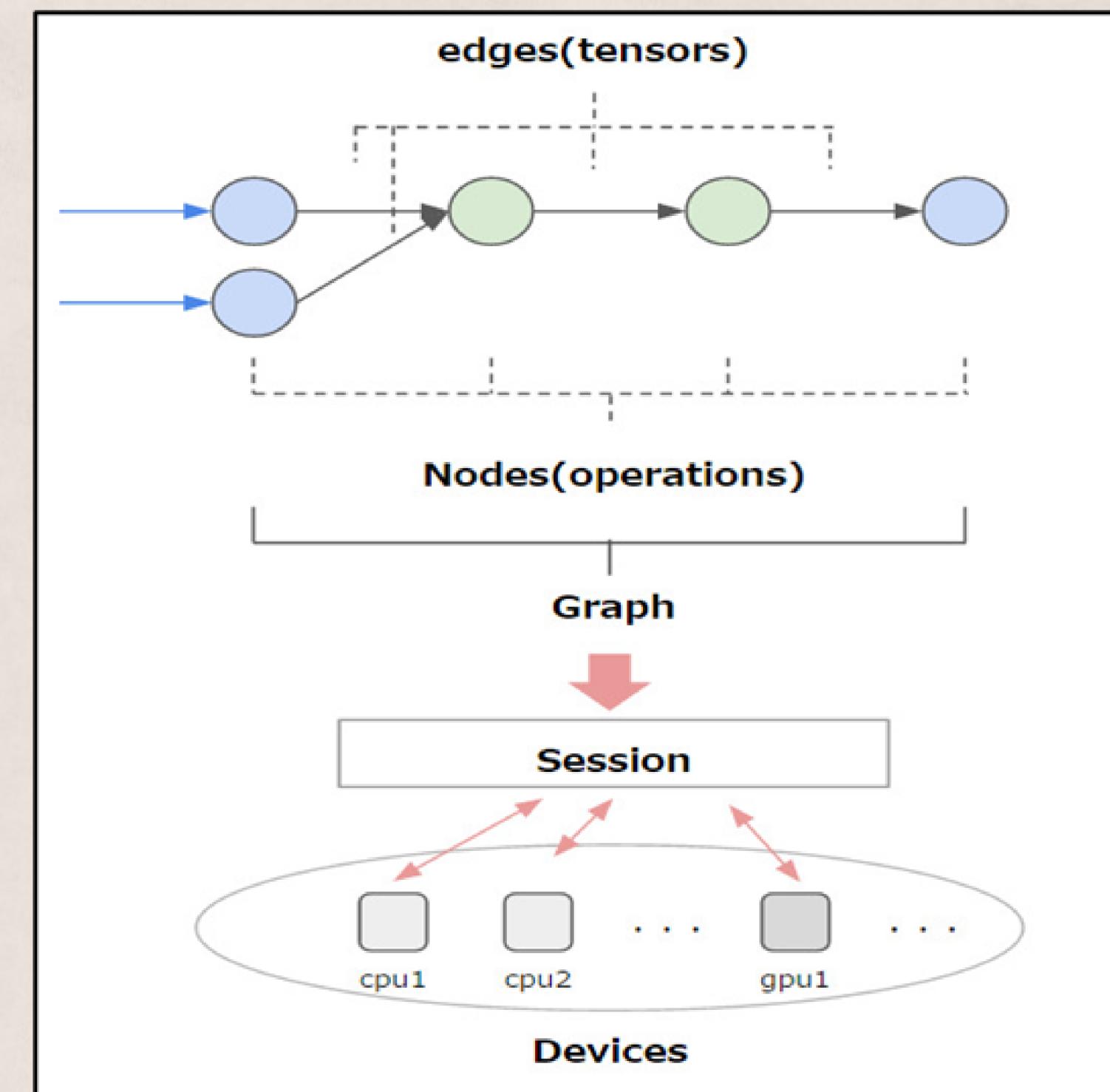
Kiến trúc Tensorflow

Kiến trúc hoạt động của Tensorflow được chia thành 3 phần chính là:

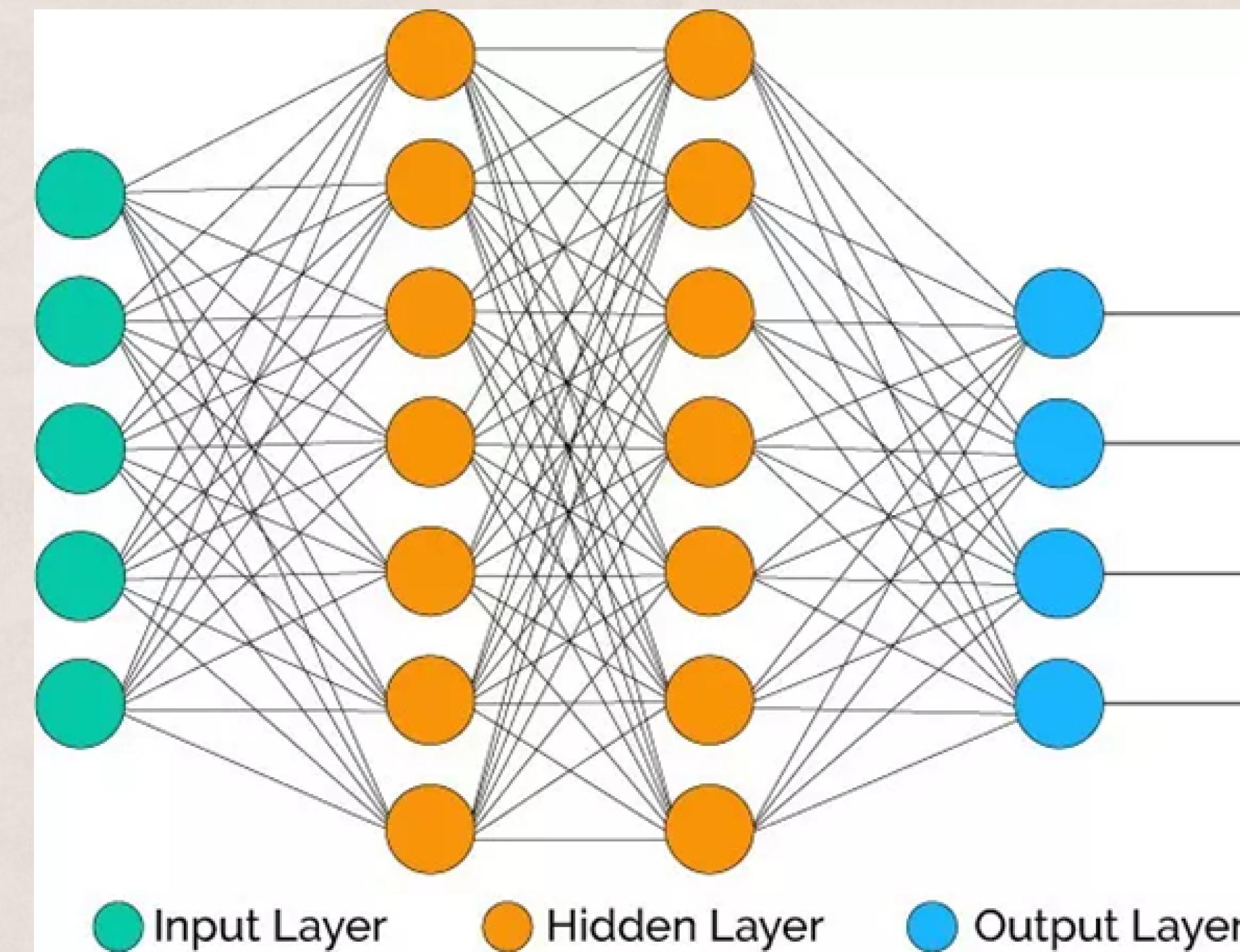
- *Tiền xử lý dữ liệu.*
- *Dựng model.*
- *Train và ước tính model cụ thể.*

Thành phần chính

Gồm 2 thành phần chính là *Tensor* và *Graph*.



Mô hình ANN



ALGORITHM MODELS

...ing brought back to life anywhere

Chia tập dữ liệu

Xóa cột "Exited" và chia tập dữ liệu thành 2 phần: train and test.

```
[ ] X = df2.drop('Exited',axis = 1)
y = df2['Exited']

[ ] from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2)
```

Chia tập dữ liệu

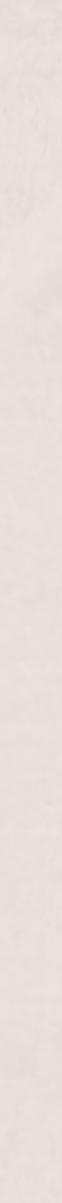
Train

80 %

Test

20 %

random_state = 42



Sử dụng Tensorflow và Keras

```
[ ] import tensorflow as tf
from tensorflow import keras

model = keras.Sequential([
    keras.layers.Dense(32,input_shape = (13,),activation='relu'),
    keras.layers.Dense(16,activation='relu'),
    keras.layers.Dense(1,activation='sigmoid')
])
model.compile(optimizer='adam',loss = 'binary_crossentropy',metrics=[ 'accuracy'])
```

Training Models

Accuracy: 86.85%

```
[ ] model.evaluate(X_test,y_test)

63/63 [=====] - 0s 2ms/step - loss: 0.3282 - accuracy: 0.8685
[0.3281884491443634, 0.8684999942779541]
```

Predict model

Dự đoán trên Test Set và sử dụng Ma trận nhầm lẫn và Báo cáo phân loại để đánh giá hiệu suất của mô hình

```
[ ] yp = model.predict(X_test)

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

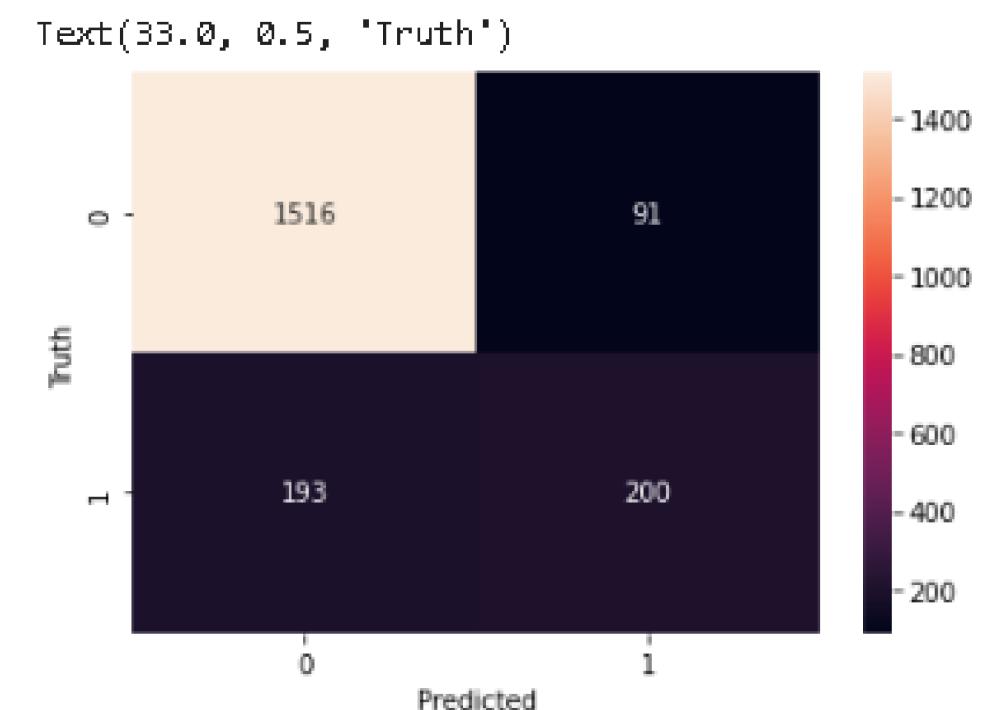
63/63 [=====] - 0s 1ms/step
```

Predict model

```
▶ from sklearn.metrics import confusion_matrix,classification_report  
print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
0	0.89	0.94	0.91	1607
1	0.69	0.51	0.58	393
accuracy			0.86	2000
macro avg	0.79	0.73	0.75	2000
weighted avg	0.85	0.86	0.85	2000

```
[ ] import seaborn as sn  
cm = tf.math.confusion_matrix(labels = y_test, predictions = y_pred)  
sn.heatmap(cm,annot = True,fmt ='d')  
plt.xlabel('Predicted')  
plt.ylabel('Truth')
```





ng brought back to life anyw

KẾT LUẬN

REFERENCES

- [1] *Slide môn Cloud Computing trường đại học CNTT - DHQG TPHCM.*
- [2] *What is Precision, Recall and F1-score? - The Conscious's notes (wordpress.com)*
- [3] *Tổng quan về Artificial Neural Network (viblo.asia)*
- [4] *What is Ordinal Data? [Definition, Analysis & Examples] (careerfoundry.com)*
- [5] *Tổng quan về Neural Network(mạng Nơ Ron nhân tạo) là gì? (itnavi.com.vn)*
- [6] *Video youtube:*
- [7] *Dataset: Bank Customer Churn Dataset | Kaggle*



BANK

THANKS FOR
WATCHING