

In [32]: #Importing necessary Libraries

```
import numpy as np
import pandas as pd
import tensorflow as tf
```

In [33]: #Loading Dataset

```
data = pd.read_csv("loan_data_set2.csv")
data.info()
data.describe()
data.shape
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 614 entries, 0 to 613
Data columns (total 12 columns):
 #   Column           Non-Null Count  Dtype  
 ---  --  
 0   Loa0_ID          614 non-null    object  
 1   Ge0der           601 non-null    float64 
 2   Married          611 non-null    float64 
 3   Depe0de0ts       599 non-null    object  
 4   Self_Emplo1ed   582 non-null    float64 
 5   Applica0tI0come 614 non-null    int64   
 6   Coapplica0tI0come 614 non-null    float64 
 7   Loa0Amou0t       592 non-null    float64 
 8   Loa0_Amou0t_Term 600 non-null    float64 
 9   Credit_Histor1  564 non-null    float64 
 10  Propert1_Area   614 non-null    int64   
 11  Loa0_Status      614 non-null    int64  
dtypes: float64(7), int64(3), object(2)
memory usage: 57.7+ KB
```

Out[33]: (614, 12)

In [34]: data

Out[34]:

	Loa0_ID	Ge0der	Married	Depe0de0ts	Self_Emplo1ed	Applica0tl0come	Coapplica0tl0come	L
0	LP001002	1.0	0.0	0	0.0	5849		0.0
1	LP001003	1.0	1.0	1	0.0	4583		1508.0
2	LP001005	1.0	1.0	0	1.0	3000		0.0
3	LP001006	1.0	1.0	0	0.0	2583		2358.0
4	LP001008	1.0	0.0	0	0.0	6000		0.0
...
609	LP002978	0.0	0.0	0	0.0	2900		0.0
610	LP002979	1.0	1.0	3+	0.0	4106		0.0
611	LP002983	1.0	1.0	1	0.0	8072		240.0
612	LP002984	1.0	1.0	2	0.0	7583		0.0
613	LP002990	0.0	0.0	0	1.0	4583		0.0

614 rows × 12 columns



```
In [35]: #Generating Matrix of Features
X = data.iloc[:,2:-1].values
print(X)
```

```
[[0.0 '0' 0.0 ... 360.0 1.0 1]
 [1.0 '1' 0.0 ... 360.0 1.0 3]
 [1.0 '0' 1.0 ... 360.0 1.0 1]
 ...
 [1.0 '1' 0.0 ... 360.0 1.0 1]
 [1.0 '2' 0.0 ... 360.0 1.0 1]
 [0.0 '0' 1.0 ... 360.0 0.0 2]]
```

```
In [36]: data.replace(to_replace="Yes",value="1")
data.replace(to_replace="No",value="0")

print(X)
```

```
[[0.0 '0' 0.0 ... 360.0 1.0 1]
 [1.0 '1' 0.0 ... 360.0 1.0 3]
 [1.0 '0' 1.0 ... 360.0 1.0 1]
 ...
 [1.0 '1' 0.0 ... 360.0 1.0 1]
 [1.0 '2' 0.0 ... 360.0 1.0 1]
 [0.0 '0' 1.0 ... 360.0 0.0 2]]
```

```
In [37]: #Generating Dependent Variable Vectors
Y = data.iloc[:,-1].values
print(Y)
```

```
[1 0 1 1 1 1 0 1 0 1 1 1 0 1 1 1 0 0 1 0 1 0 0 0 1 1 1 0 1 0 0 0 1 0 1 1 1 0
1 1 1 0 1 1 1 1 1 0 1 1 1 0 0 0 1 1 0 1 1 1 1 0 0 0 0 0 1 1 0 1 1 1 0
1 0 0 0 0 1 1 1 0 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 0 0 1 1
1 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 0 1 0 0 0 1 1 1 1 1 1 1 0 0 1 1
0 1 0 1 0 0 1 1 1 1 1 1 1 0 0 1 1 1 0 1 0 1 1 1 0 1 0 1 1 0 1 0 1 1 0 0 0 1 0 1
1 0 1 1 1 1 0 0 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 0 0 0 1 1 1 1 1 0 0 1 0 0 0 1 1 1 1 1 1
1 1 1 0 0 1 1 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 0 0 1 1 0 1 0 0 0
0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 0 1 0 1 1 1 1 0 1 0 1 1 1
1 1 0 0 0 1 1 1 1 0 1 0 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1
1 1 1 1 1 0 1 0 0 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 0 1 0 1 1 1 1 1 0 0 0 1 0 1 1 1 1
1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 0 0 1 1 1 0 1 1 1 1 1 0 1 1 1
1 0 0 0 1 0 1 0 1 0 0 1 1 1 1 0 1 1 1 1 0 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1
1 1 1 0 0 0 0 1 0 1 1 1 1 0 1 0 1 1 1 1 0 1 0 1 1 1 1 0 1 0 1 1 1 1 1 0 1 0 1 1 1 1 0 1 1
1 1 1 1 1 0 0 1 0 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 0 1 1 1 0 0 1 1
0 0 1 1 1 1 0 1 1 1 1 1 1 0 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 0 1 0 1 1 1 0 0
1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 0 1 0
1 1 1 1 1 0 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 0]
```

In [50]: `data`

	Loa0_ID	Ge0der	Married	Depe0de0ts	Self_Emplo1ed	Applica0tl0come	Coapplica0tl0come	L
0	LP001002	1.0	0.0	0	0.0	5849		0.0
1	LP001003	1.0	1.0	1	0.0	4583		1508.0
2	LP001005	1.0	1.0	0	1.0	3000		0.0
3	LP001006	1.0	1.0	0	0.0	2583		2358.0
4	LP001008	1.0	0.0	0	0.0	6000		0.0
...
609	LP002978	0.0	0.0	0	0.0	2900		0.0
610	LP002979	1.0	1.0	3+	0.0	4106		0.0
611	LP002983	1.0	1.0	1	0.0	8072		240.0
612	LP002984	1.0	1.0	2	0.0	7583		0.0
613	LP002990	0.0	0.0	0	1.0	4583		0.0

614 rows × 12 columns

In [38]: `#Encoding Categorical Variable Gender`
`from sklearn.preprocessing import LabelEncoder`
`LE1 = LabelEncoder()`
`X[:,2] = np.array(LE1.fit_transform(X[:,2]))`

In [39]: `#Encoding Categorical variable Geography`
`from sklearn.compose import ColumnTransformer`
`from sklearn.preprocessing import OneHotEncoder`
`ct = ColumnTransformer(transformers=[('encoder', OneHotEncoder(), [1])], remainder="passthrough")`
`X = np.array(ct.fit_transform(X))`

In [40]: `#Splitting dataset into training and testing dataset`
`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,random_state=0)`

```
In [41]: #Performing Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)

In [42]: #Initialising ANN
ann = tf.keras.models.Sequential()

In [43]: #Adding First Hidden Layer
ann.add(tf.keras.layers.Dense(units=10,activation="relu"))

In [44]: #Adding Second Hidden Layer
ann.add(tf.keras.layers.Dense(units=6,activation="relu"))

In [45]: #Adding Output Layer
ann.add(tf.keras.layers.Dense(units=1,activation="sigmoid"))

In [46]: #Compiling ANN
ann.compile(optimizer="adam", loss="binary_crossentropy", metrics=['accuracy'])

In [47]: #Fitting ANN
ann.fit(X_train,Y_train,batch_size=32,epochs = 100)
```

Epoch 1/100
16/16 [=====] - 1s 1ms/step - loss: nan - accuracy: 0.3442
Epoch 2/100
16/16 [=====] - 0s 997us/step - loss: nan - accuracy: 0.3238
Epoch 3/100
16/16 [=====] - 0s 931us/step - loss: nan - accuracy: 0.3238
Epoch 4/100
16/16 [=====] - 0s 931us/step - loss: nan - accuracy: 0.3238
Epoch 5/100
16/16 [=====] - 0s 931us/step - loss: nan - accuracy: 0.3238
Epoch 6/100
16/16 [=====] - 0s 997us/step - loss: nan - accuracy: 0.3238
Epoch 7/100
16/16 [=====] - 0s 931us/step - loss: nan - accuracy: 0.3238
Epoch 8/100
16/16 [=====] - 0s 931us/step - loss: nan - accuracy: 0.3238
Epoch 9/100
16/16 [=====] - 0s 865us/step - loss: nan - accuracy: 0.3238
Epoch 10/100
16/16 [=====] - 0s 864us/step - loss: nan - accuracy: 0.3238
Epoch 11/100
16/16 [=====] - 0s 931us/step - loss: nan - accuracy: 0.3238
Epoch 12/100
16/16 [=====] - 0s 931us/step - loss: nan - accuracy: 0.3238
Epoch 13/100
16/16 [=====] - 0s 864us/step - loss: nan - accuracy: 0.3238
Epoch 14/100
16/16 [=====] - 0s 931us/step - loss: nan - accuracy: 0.3238
Epoch 15/100
16/16 [=====] - 0s 931us/step - loss: nan - accuracy: 0.3238
Epoch 16/100
16/16 [=====] - 0s 864us/step - loss: nan - accuracy: 0.3238
Epoch 17/100
16/16 [=====] - 0s 864us/step - loss: nan - accuracy: 0.3238
Epoch 18/100
16/16 [=====] - 0s 997us/step - loss: nan - accuracy: 0.3238
Epoch 19/100
16/16 [=====] - 0s 931us/step - loss: nan - accuracy: 0.3238
Epoch 20/100
16/16 [=====] - 0s 931us/step - loss: nan - accuracy: 0.3238
Epoch 21/100
16/16 [=====] - 0s 931us/step - loss: nan - accuracy: 0.3238
Epoch 22/100
16/16 [=====] - 0s 931us/step - loss: nan - accuracy: 0.3238
Epoch 23/100
16/16 [=====] - 0s 931us/step - loss: nan - accuracy: 0.3238
Epoch 24/100
16/16 [=====] - 0s 932us/step - loss: nan - accuracy: 0.3238
Epoch 25/100
16/16 [=====] - 0s 931us/step - loss: nan - accuracy: 0.3238
Epoch 26/100
16/16 [=====] - 0s 864us/step - loss: nan - accuracy: 0.3238
Epoch 27/100
16/16 [=====] - 0s 864us/step - loss: nan - accuracy: 0.3238
Epoch 28/100
16/16 [=====] - 0s 931us/step - loss: nan - accuracy: 0.3238
Epoch 29/100
16/16 [=====] - 0s 931us/step - loss: nan - accuracy: 0.3238
Epoch 30/100
16/16 [=====] - 0s 931us/step - loss: nan - accuracy: 0.3238

Epoch 31/100
16/16 [=====] - 0s 997us/step - loss: nan - accuracy: 0.3238
Epoch 32/100
16/16 [=====] - 0s 864us/step - loss: nan - accuracy: 0.3238
Epoch 33/100
16/16 [=====] - 0s 931us/step - loss: nan - accuracy: 0.3238
Epoch 34/100
16/16 [=====] - 0s 997us/step - loss: nan - accuracy: 0.3238
Epoch 35/100
16/16 [=====] - 0s 931us/step - loss: nan - accuracy: 0.3238
Epoch 36/100
16/16 [=====] - 0s 931us/step - loss: nan - accuracy: 0.3238
Epoch 37/100
16/16 [=====] - 0s 931us/step - loss: nan - accuracy: 0.3238
Epoch 38/100
16/16 [=====] - 0s 931us/step - loss: nan - accuracy: 0.3238
Epoch 39/100
16/16 [=====] - 0s 930us/step - loss: nan - accuracy: 0.3238
Epoch 40/100
16/16 [=====] - 0s 864us/step - loss: nan - accuracy: 0.3238
Epoch 41/100
16/16 [=====] - 0s 931us/step - loss: nan - accuracy: 0.3238
Epoch 42/100
16/16 [=====] - 0s 864us/step - loss: nan - accuracy: 0.3238
Epoch 43/100
16/16 [=====] - 0s 932us/step - loss: nan - accuracy: 0.3238
Epoch 44/100
16/16 [=====] - 0s 1ms/step - loss: nan - accuracy: 0.3238
Epoch 45/100
16/16 [=====] - 0s 931us/step - loss: nan - accuracy: 0.3238
Epoch 46/100
16/16 [=====] - 0s 930us/step - loss: nan - accuracy: 0.3238
Epoch 47/100
16/16 [=====] - 0s 864us/step - loss: nan - accuracy: 0.3238
Epoch 48/100
16/16 [=====] - 0s 931us/step - loss: nan - accuracy: 0.3238
Epoch 49/100
16/16 [=====] - 0s 931us/step - loss: nan - accuracy: 0.3238
Epoch 50/100
16/16 [=====] - 0s 931us/step - loss: nan - accuracy: 0.3238
Epoch 51/100
16/16 [=====] - 0s 931us/step - loss: nan - accuracy: 0.3238
Epoch 52/100
16/16 [=====] - 0s 1ms/step - loss: nan - accuracy: 0.3238
Epoch 53/100
16/16 [=====] - 0s 997us/step - loss: nan - accuracy: 0.3238
Epoch 54/100
16/16 [=====] - 0s 863us/step - loss: nan - accuracy: 0.3238
Epoch 55/100
16/16 [=====] - 0s 864us/step - loss: nan - accuracy: 0.3238
Epoch 56/100
16/16 [=====] - 0s 863us/step - loss: nan - accuracy: 0.3238
Epoch 57/100
16/16 [=====] - 0s 864us/step - loss: nan - accuracy: 0.3238
Epoch 58/100
16/16 [=====] - 0s 864us/step - loss: nan - accuracy: 0.3238
Epoch 59/100
16/16 [=====] - 0s 931us/step - loss: nan - accuracy: 0.3238
Epoch 60/100
16/16 [=====] - 0s 864us/step - loss: nan - accuracy: 0.3238

Epoch 61/100
16/16 [=====] - 0s 931us/step - loss: nan - accuracy: 0.3238
Epoch 62/100
16/16 [=====] - 0s 864us/step - loss: nan - accuracy: 0.3238
Epoch 63/100
16/16 [=====] - 0s 864us/step - loss: nan - accuracy: 0.3238
Epoch 64/100
16/16 [=====] - 0s 864us/step - loss: nan - accuracy: 0.3238
Epoch 65/100
16/16 [=====] - 0s 863us/step - loss: nan - accuracy: 0.3238
Epoch 66/100
16/16 [=====] - 0s 997us/step - loss: nan - accuracy: 0.3238
Epoch 67/100
16/16 [=====] - 0s 997us/step - loss: nan - accuracy: 0.3238
Epoch 68/100
16/16 [=====] - 0s 931us/step - loss: nan - accuracy: 0.3238
Epoch 69/100
16/16 [=====] - 0s 931us/step - loss: nan - accuracy: 0.3238
Epoch 70/100
16/16 [=====] - 0s 864us/step - loss: nan - accuracy: 0.3238
Epoch 71/100
16/16 [=====] - 0s 931us/step - loss: nan - accuracy: 0.3238
Epoch 72/100
16/16 [=====] - 0s 864us/step - loss: nan - accuracy: 0.3238
Epoch 73/100
16/16 [=====] - 0s 931us/step - loss: nan - accuracy: 0.3238
Epoch 74/100
16/16 [=====] - 0s 997us/step - loss: nan - accuracy: 0.3238
Epoch 75/100
16/16 [=====] - 0s 864us/step - loss: nan - accuracy: 0.3238
Epoch 76/100
16/16 [=====] - 0s 863us/step - loss: nan - accuracy: 0.3238
Epoch 77/100
16/16 [=====] - 0s 864us/step - loss: nan - accuracy: 0.3238
Epoch 78/100
16/16 [=====] - 0s 864us/step - loss: nan - accuracy: 0.3238
Epoch 79/100
16/16 [=====] - 0s 864us/step - loss: nan - accuracy: 0.3238
Epoch 80/100
16/16 [=====] - 0s 864us/step - loss: nan - accuracy: 0.3238
Epoch 81/100
16/16 [=====] - 0s 864us/step - loss: nan - accuracy: 0.3238
Epoch 82/100
16/16 [=====] - 0s 864us/step - loss: nan - accuracy: 0.3238
Epoch 84/100
16/16 [=====] - 0s 864us/step - loss: nan - accuracy: 0.3238
Epoch 85/100
16/16 [=====] - 0s 864us/step - loss: nan - accuracy: 0.3238
Epoch 86/100
16/16 [=====] - 0s 1ms/step - loss: nan - accuracy: 0.3238
Epoch 87/100
16/16 [=====] - 0s 864us/step - loss: nan - accuracy: 0.3238
Epoch 88/100
16/16 [=====] - 0s 864us/step - loss: nan - accuracy: 0.3238
Epoch 89/100
16/16 [=====] - 0s 798us/step - loss: nan - accuracy: 0.3238
Epoch 90/100
16/16 [=====] - 0s 864us/step - loss: nan - accuracy: 0.3238

```
Epoch 91/100
16/16 [=====] - 0s 864us/step - loss: nan - accuracy: 0.3238
Epoch 92/100
16/16 [=====] - 0s 864us/step - loss: nan - accuracy: 0.3238
Epoch 93/100
16/16 [=====] - 0s 931us/step - loss: nan - accuracy: 0.3238
Epoch 94/100
16/16 [=====] - 0s 864us/step - loss: nan - accuracy: 0.3238
Epoch 95/100
16/16 [=====] - 0s 864us/step - loss: nan - accuracy: 0.3238
Epoch 96/100
16/16 [=====] - 0s 931us/step - loss: nan - accuracy: 0.3238
Epoch 97/100
16/16 [=====] - 0s 931us/step - loss: nan - accuracy: 0.3238
Epoch 98/100
16/16 [=====] - 0s 864us/step - loss: nan - accuracy: 0.3238
Epoch 99/100
16/16 [=====] - 0s 864us/step - loss: nan - accuracy: 0.3238
Epoch 100/100
16/16 [=====] - 0s 864us/step - loss: nan - accuracy: 0.3238
Out[47]: <keras.src.callbacks.History at 0x1dedb717d90>
```

```
In [55]: #Predicting result for Single Observation
print(ann.predict(sc.transform([[1, 0, 0, 0, 5849, 0, 0, 360, 1, 1, 1, 1, 1]])) > 0.5)
1/1 [=====] - 0s 58ms/step
[[False]]
```

```
In [ ]:
```